

AD-A238 459



DTIC
FCTE
JUL 23 1991

A VISION SYSTEM MODEL

DISSERTATION

Erik J. Fretheim
Captain, US Army
AFIT/DS/ENG/91-1

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA 22161

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DS/ENG/91-1

1

DTIC
ELECTE
JUL 23 1991
S D

A VISION SYSTEM MODEL

DISSERTATION

Erik J. Fretheim
Captain, US Army
AFIT/DS/ENG/91-1

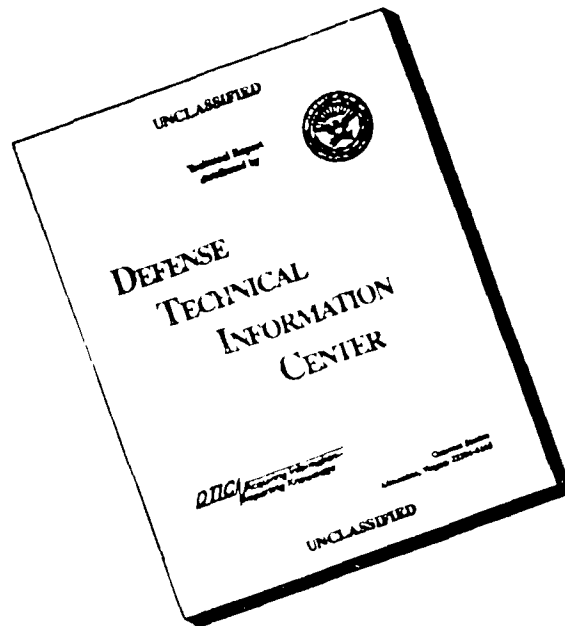
Approved for public release; distribution unlimited.

91 1 145

91-05769



DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

A VISION SYSTEM MODEL

Erik J. Fretheim, B.S., M.S.E.E.

Captain, US Army

Approved:



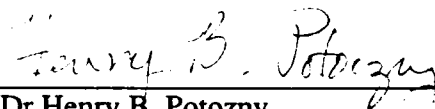
18 May 91

Dr Matthew Kabrisky, Chairman



17 MAY 91

Dr Frank M. Brown



May 17, 1991

Dr Henry B. Potozny

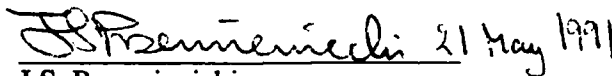


20 MAY 1991

Dr Steven K. Rogers

Approved By	
J	
Title	
Date	
By	
Date	
Reviewed By	
Date	
Dist	Avail. and Spec.
A-1	

Accepted:



J.S. Przemienicki
Senior Dean



REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188	
1. AGENCY USE ONLY (Leave blank)			
2. REPORT DATE June 1991		3. REPORT TYPE AND DATES COVERED PhD Dissertation	
4. TITLE AND SUBTITLE A VISION SYSTEM MODEL		5. FUNDING NUMBERS	
6. AUTHOR(S) Erik J. Fretheim, Captain, U.S. Army		7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) School of Engineering Air Force Institute of Technology (AU) Wright-Patterson AFB, OH 45433	
8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENG/91-1		9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)	
10. SPONSORING MONITORING AGENCY REPORT NUMBER		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This dissertation provides four major contributions to the field of vision research. The first contribution is a general vision system model. The model, which blends biological as well as technological methods into a coherent approach to vision, will provide a basis for implementing vision systems. The second contribution of this dissertation is to demonstrate particular implementations of portions of the model. These implementations will include methods for using Gabor wavelets in edge detection, in preprocessing images for use as feature vectors in backpropagation neural networks, and as basis functions in a recognition/reconstruction network, as well as methods for integrating color into a vision system. The third major contribution is an investigation of attentional mechanisms using a two-part model with Gabor filters as a base attentional indicator. The second part of the model, the search mechanism, is only briefly studied. The final contribution is a description of an actual vision system for the analysis of VLSI circuits in terms of the general vision system model. This system provides a means of obtaining a logical circuit description from an actual physical circuit.			
14. SUBJECT TERMS Computer Vision; Pattern recognition Image processing; Circuit inspection		15. NUMBER OF PAGES 271	
16. PRICE CODE		17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
20. LIMITATION OF ABSTRACT			

A VISION SYSTEM MODEL

DISSERTATION

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Erik J. Fretheim, B.S., M.S.E.E.

Captain, US Army

June 1991

Approved for public release; distribution unlimited.

Acknowledgements

I have received a lot of support in the course of this work, not only from my family and instructors, but also from my fellow students. I would like to especially acknowledge those who have worked with me on the Reverse Engineering project: Capt Mike Meuller; Capt Gene Leano; Cpt Rick Hayden; Capt. Mike Shoop; Capt Eric Augustus; Capt John Querns; Lt Jim Lee; Capt. Bruce Plourde; and Lt Rod Taylor; also Capt Ralph St John, who is working on attention (not at). This work could not have been completed without their assistance and forbearance.

I would thank Dr Kabrisky for all of his support. I guess I own him a pound of two of *Gevalia*.

Finally, and most importantly, I would like to thank my family for all of their patience, support and understanding.

TABLE OF CONTENTS

Acknowledgements	ii
List of Figures	ix
List of Tables	xii
Abstract	xiii

Chapter 1: Introduction

1.1 Background	1
1.2 Teleological Functionalism	5
1.3 Problem Statement	7
1.4 Approach	8

Chapter 2: The General Vision System Model

2.1 Overview of the Model	10
2.2 The Model's World Picture and World View	13
2.3 The Control System	17
2.3.1 Maintaining Goals	18
2.3.2 Attention Mechanisms	19
2.3.3 Long Term Storage	21
2.3.4 Feedback	22
2.4 The Sensors and Their Transforms	23
2.5 Input transforms	26
2.6 The Super-Conscious	26

Chapter 3: Mechanisms for a Vision System

TABLE OF CONTENTS (continued)

3.1 Wavelet Models	31
3.1.1 Gabor Wavelets	35
3.1.2 Gabor Wavelets and the Visual Cortex	36
3.2 Psychological Implications of Gabor Wavelets	42
3.2.1 The Simultaneous Contrast Bar	43
3.2.2 The Spreading Effect or Assimilation	47
3.2.3 The Contrast Sensitivity Function	48
3.2.4 The Muller-Lyer, Ponzo and Other Illusions	50
3.3 Applying Gabor Wavelets to the Vision System Model	58
3.4 Edge Location	59
3.5 Object Identification	64
3.5.1 The Pseudo-neocognitron	67
3.6 Attention Mechanisms in Visual Systems	72
3.6.1 The Superior Colliculus	72
3.6.2 The Posterior Parietal Cortex	74
3.6.3 The Gabor Transform as an Attention Mechanism	76
3.6.4 Experimental Design	82
3.6.5 Experimental Setup	84
3.6.6 Experimental Procedures	86
3.6.7 Data Analysis	87
3.7 Light, Color and Lasers	99
3.7.1 Monochromaticity	100
3.7.2 Multi-Colors	107
3.7.3 Opponent Colors	113
 Chapter 4: The AFIT Reverse Engineering System	
4.1 Background	117
4.2 Hardware Base	123

TABLE OF CONTENTS (continued)

4.3 CHIP	126
4.4 Control	133
4.5 The Sensors and Input Processes	135
4.5.1 Lasers, Filters and Color	135
4.5.2 Color Manipulations	140
4.5.3 Edge and Region Locations	145
4.5.4 Contact Finding	146
4.6 Internal Transforms	152
 Chapter 5: Conclusions and Recommendations	
5.1 Conclusions	154
5.1.1 The Vision System Model	154
5.1.2 Gabor Filters	155
5.1.3 The Color Model	157
5.2 Recommendations	157
5.2.1 Attention and Search Strategies	158
5.2.2 Pseudo-neocognitron	159
5.2.3 Reverse Engineering	160
5.2.4 Use of the CHIP System	162
 References	
 Appendix A: Vision	
1 Introduction	2
2 Defining Vision	6
2.1 Seeing and Recognition	7

TABLE OF CONTENTS (continued)

2.2 Components of Vision	8
2.3 Two Kinds of Seeing	11
2.4 External Control of Vision Systems	13
3 Goals of a Vision System Model	15
3.1 Adaptability	15
3.2 An Improved System	16
3.3 Realizable and Realistic	17
3.4 Human/Machine Factors Considerations	19
4 References	19

Appendix B: Testchip

1 Introduction	2
2 The Circuit Section	3
3 The Large-Feature Section	3
4 Summary	4
5 References	4

Appendix C: The CHIP System User's Guide

1 Introduction	2
2 The Hardware Base	3

TABLE OF CONTENTS (continued)

3 System Startup	5
4 CHIP Windows	6
4.1 The MITAS Window	7
4.1.1 Initializing the MITAS controller	8
4.1.2 MITAS Buttons	10
4.1.3 MITAS Commands	10
4.2 The CLIPS Window	11
4.3 The CHIP Window	13
4.3.1 CHIP Menu Commands	15
4.3.2 The CHIP Menu Grammar	15
4.3.3 Reading Chip Menu Commands From a File	17
4.3.4 Image Display, Creation and Storage Commands	18
4.3.5 Template Manipulations	21
4.3.6 Area Movements	22
4.3.7 Layer and Block Extractions	22
4.3.8 Laser Operations	25
4.3.9 The RCS Simulator	25
4.3.10 Neocognitrons	26
4.3.11 Density Domain Commands	28
4.3.12 Edge Detection	29
4.3.13 External Programs	29
4.3.14 Localized Transforms	29
4.3.15 Global Transforms	30
4.3.16 Logical Operations	31
4.3.17 Parameter Settings	32
4.3.18 Quick Reference for CHIP Menu Commands	33
5 References:	45
 Appendix D: The Chip System Programmer's Manual	
1 Introduction	2

TABLE OF CONTENTS (continued)

2 The Hardware Base	3
3 The Programming Model	4
4 Makefile System	5
5 Adding Routines	7
5.1 Hardware-Specific Programs	9
5.2 Adding to the Menus	10
5.3 Adding to the User's Guide	10
5.4 Adding to the Windows	11
5.5 Adding CLIPS routines	12
5.5.1 CLIPS Rule Files	12
6 References:	13
 Appendix E: Subject Requirements and Consent Forms	
1 Subject Requirements	2
2 Consent Form	3
3 Addendum	4

Vita

List of Figures

Figure 1: Relationship Between the Vision System Model and the System for Reverse-Engineering VLSI	3
Figure 2: The Vision System Model	10
Figure 3: A Deceptive Close Up	18
Figure 4: Adaptation Curves for the Eye Under Varied Lighting Conditions [Goldstein, p. 81]	24
Figure 5: Wavelet Samples: A) Modulation functions B) Envelopes C) Wavelets	32
Figure 6: Self-Similar (a) and Affine (b) Wavelets	33
Figure 7: Annulus of Coverage for a Set of Equal-Sized Wavelets [Mallat 89a, p. 35]	34
Figure 8: Construction of a Gabor Wavelet [Jones and Palmer, p. 1235]	35
Figure 9: Typical Receptive Fields of Cortical Cells	37
Figure 10: Comparison of Cat Cortex Simple Cells to Gabor Filters [Jones and Palmer, p. 1238]	39
Figure 11: Comparison of Gabor Filters and Cat Cortex Simple Cells With Error [Jones and Palmer, p. 1242]	40
Figure 12: Complex Cells - 2-bar Interactions [Emerson et al, 90, p. 9]	41
Figure 13: 3-d Filters [Citron et al., p. 184]	42
Figure 14: Simultaneous Contrast Bar	43
Figure 15: Constant Intensity Bar After Gabor Filtering to Enhance Edges	45
Figure 16: Simultaneous Contrast Bar After Filtering to Change the Bar Intensities	46
Figure 17: Combined Wavelet Filtering of the Constant Intensity Bar Image	46
Figure 18: Assimilation of Closely Spaced Lines:	47
Figure 19: Logarithmic Sine-Wave Grating	49
Figure 20: Human Sensitivity Curve to Sine-Wave Grating [Goldstein, p. 163]	50
Figure 21: Gabor Filtered Sine-wave Grating	50
Figure 22: The Muller-Lyer (a) and Ponzo (b) Illusions	51
Figure 23: Modified Muller-Lyer Image with Illusion Reversed	52
Figure 24: Gabor Filtered Muller-Lyer Illusion	54
Figure 25: Two filtered Ponzo Illusions	54
Figure 26: Aspects of the Poggendorff Illusion	55
Figure 27: The Zollner Illusion	56
Figure 28: The Wundt Illusion	57
Figure 29: Gradient Flow Diagram	62
Figure 30: Image Gradients Coded as Difference of Horizontal and Vertical Gabor Correlations	63
Figure 31: Edge Types: A) Roof Edges. B) Step Edges. C) Slope Edges.	63

List of Figures (continued)

Figure 32: Structure of a Neocognitron [Fukushima, p. 4986]	68
Figure 33: Pseudo-Neocognitron Input and Output	71
Figure 34: Superior Colliculus in the Mid Section of the Brain Stem	73
Figure 35: Typical Eye Movements When Viewing a Face	77
Figure 36: Correlation of Gabor Filters with an Image of a Face	78
Figure 37: Combined Correlations of 0 and 90 Degree Gabor Filters with the Couch Snake Image	80
Figure 38: Eye Movements Elicited by Different Verbal Prompts [Yarbus, p. 174]	81
Figure 39: Corneal and Pupal Reflections as Seen by CCD Camera	84
Figure 40: Example of Expected Eye Jitter During a Single Fixation	88
Figure 41: Fixation Circles Plotted on Eye Movement Points	90
Figure 42: Distribution of Test Points within the Foveal Field	91
Figure 43: Sample of Gabor Filters - Size 18	92
Figure 44: Histograms of the Relative Frequency for Gabor Correlation Values for a Representative Image: A - Random Points. B-G - Subjects	95
Figure 45: Box Image	96
Figure 46: Histograms of Correlation Distributions for Box Image	97
Figure 47: Microscopic vrs. Macroscopic Interactions with the Electro-Magnetic Spectrum	100
Figure 48: Factors Influencing Transmitted Light	102
Figure 49: Reflection Intensity Versus Film Thickness	106
Figure 50: Human Visual Frequency Response Curves	108
Figure 51: 1931 CIE Chromaticity Diagram [Augustus, p. 27]	109
Figure 52: System to Provide Color Response Equivalent to the Eye	113
Figure 53: Red-Green and Blue-Yellow Response Curves [Goldstein, p. 125]	114
Figure 54: Plot of Opponent Values	116
Figure 55: Typical CAD cycle	121
Figure 56: SUN IV System Configuration	124
Figure 57: SUN III System Configuration	125
Figure 58: Cell Structure in MAGIC	128
Figure 59: Typical CLIPS rule	131
Figure 60: CLIPS Window	131
Figure 61: MITAS Window	132
Figure 62: The Chip Menu	133
Figure 63: Expected Reflectivities for White Light	138
Figure 64: Chromaticity of Silicon Dioxide on Silicon for a Thickness Range of 50- 500nm [Augustus, p. 44]	139
Figure 65: Reflection of Heavily Doped Silicon Samples: 1) $2 \times 10^{20} \text{ cm}^{-3}$ (B); 2) $5 \times 10^{19} \text{ cm}^{-3}$ (B); 3) $2 \times 10^{19} \text{ cm}^{-3}$ (As); 4) $1 \times 10^{18} \text{ cm}^{-3}$ (As) [Fistul, p. 237].	140

List of Figures (continued)

Figure 66: Results of Separation in 3 Color Space; Metal 2 and Polysilicon Shown	142
Figure 67: Red-Green and Blue-Yellow Opponent Planes	144
Figure 68: Edge Enhancement of an Image: A) Flow Diagram. B) Line Drawing	146
Figure 69: Gabor transform applied to circuit	148
Figure 70: Combined Image	149
Figure 71: Image Thresholded to Reveal Likely Contact Locations	150
Figure A-1: Limitlessly Expanding Pattern	A-4
Figure A-2: Deadly Couch Cobra Awaits Its Victim [Larson]	A-7
Figure A-3: <i>Pintos</i> by Bev Doolittle [Goldstien, p. 204]	A-8
Figure A-4: Examples of the Written Letter 'A'	A-17
Figure B-1: Testchip	B-2
Figure C-1: The Maxvideo System	C-3
Figure C-2: The MITAS Window	C-7
Figure C-3: The MITAS Controller	C-8
Figure C-4: The CLIPS Window	C-11
Figure C-5: The CHIP Window	C-13
Figure C-6: Typical CHIP command file.	C-17
Figure C-7: Quadrants on the ITEX Framebuffer	C-18
Figure C-8: Captain Linderman was in Control (A Sub-Section of the Circuit)	C-24
Figure C-9: Excerpt from a Configuration File for a Three-Layer Pseudo-Neocognitron	C-26

List of Tables

Table 1: Gabor Filter Characteristics for Fixation Analysis	91
Table 2: Results of T-Tests	93
Table 3: Estimators For The First 5 Fixations for Sample Images	99
Table 4: Results of Laser Segmentation	140
Table 5: Reduction of Contact Search Space	150
Table 6: Results of Backpropagation Classification	151
Table C-1: Image Pixrects	C-13
Table C-2: Logical and Related Operations	C-31
Table C-3: Chip System Defaults	C-32
Table D-1: Subdirectory Groupings	D-7

Abstract

This dissertation provides four major contributions to the field of vision research. The first contribution is a general vision system model. The model, which blends biological as well as technological methods into a coherent approach to vision, will provide a basis for implementing vision systems. The second contribution of this dissertation is to demonstrate particular implementations of portions of the model. These implementations will include methods for using Gabor wavelets in edge detection, in preprocessing images for use as feature vectors in backpropagation neural networks, and as basis functions in a recognition/reconstruction network, as well as methods for integrating color into a vision system. The third major contribution is an investigation of attention mechanisms using a two-part model with Gabor filters as a base attentional indicator. The second part of the model, the search mechanism, is only studied briefly. The final contribution is a description of an actual vision system for the reverse engineering of VLSI circuits in terms of the general vision system model. This system provides a means of obtaining a logical circuit description from an actual physical circuit.

A VISION SYSTEM MODEL

CHAPTER 1

Introduction

One disappointment of modern times has been the inability of the computer to recognize visual patterns. The seeming effortless with which we see and understand the world around us stands in contrast to the difficulties experienced when trying to provide machines with this same capability. When we look at what computers have done for us we see the calculator, able to add vast quantities of numbers far faster than can an individual; we see the intrinsically iterable database, able to accurately remember far more than any brain can; we see the expert system, able to make faster, more consistent (albeit stereotyped) decisions than can any human expert. And yet, when we look at computer vision we have but a few limited, special application systems. This is far from the promise of our previous examples, which suggest that we should be able to design a system which not only can see as well as we see, but one which is far better than anything we are capable of. We want a system which can distinguish features at higher resolution; we want a system which can see through the dark, smoke and other obscuring factors; we want a system which is quicker than the hand and can see where the magician has hidden the rabbit, so to speak.

1.1. Background

For my Master's Thesis, I proposed and demonstrated the feasibility of creating a system to reverse engineer VLSI circuits [Fretheim]. In trying to develop this vision system, my first approach was to look for a general vision system model to apply to the problem. Such a model

would have greatly reduced the effort required to produce such a system. Having a general vision system model would have allowed the construction of the vision system to focus on those problems peculiar to the specific task of investigating VLSI designs. However, there was no such system available and the task became much more difficult. This state of affairs is similar to that in which an automotive engineer would find himself in trying to design a car if there were no general model from which he could work. The automotive engineer knows that he needs to include wheels, a motor, a transmission, seats and a steering wheel; moreover, he knows the general relations among these components. There is no such model, however, to guide the designer of a vision system. This dissertation will address this shortfall by suggesting a general vision system model, giving specific examples of the application of the model to the problem of reverse-engineering VLSI circuits.

The usual approach to the development of vision systems has been to amalgamate a number of known and/or new image processing algorithms and synthesize them into a system to perform the desired vision tasks. In doing this each vision researcher chooses the algorithms and their combination by their "feel" and his experience. As a result the vision system design is frequently driven by the properties of the component elements rather than being guided by general principles for vision system design. The result is a lack of flexibility, the need to re-solve many problems each time a new vision system is created, and the loss of time due to the inability to focus research. In this dissertation I will propose a general theoretical structure for the creation of vision systems, and then show the application of this model to a particular problem. The theoretical model will provide the boundaries, within which the solution to the problem of building a vision system for reverse-engineering VLSI circuits lies. Not all of the functions of the model will be completely implemented within the reverse engineering system, either because the technology, or equipment, to support the functions was not available, or because the particular function fell outside of the scope of this work. The relationship between the vision system model and the

reverse engineering system is shown in Figure 1. The ties from the reverse engineering system are made through particular implementations of the functions of the model. Other vision systems can also be specified within the model; however, they may have different implementations of the same functions.

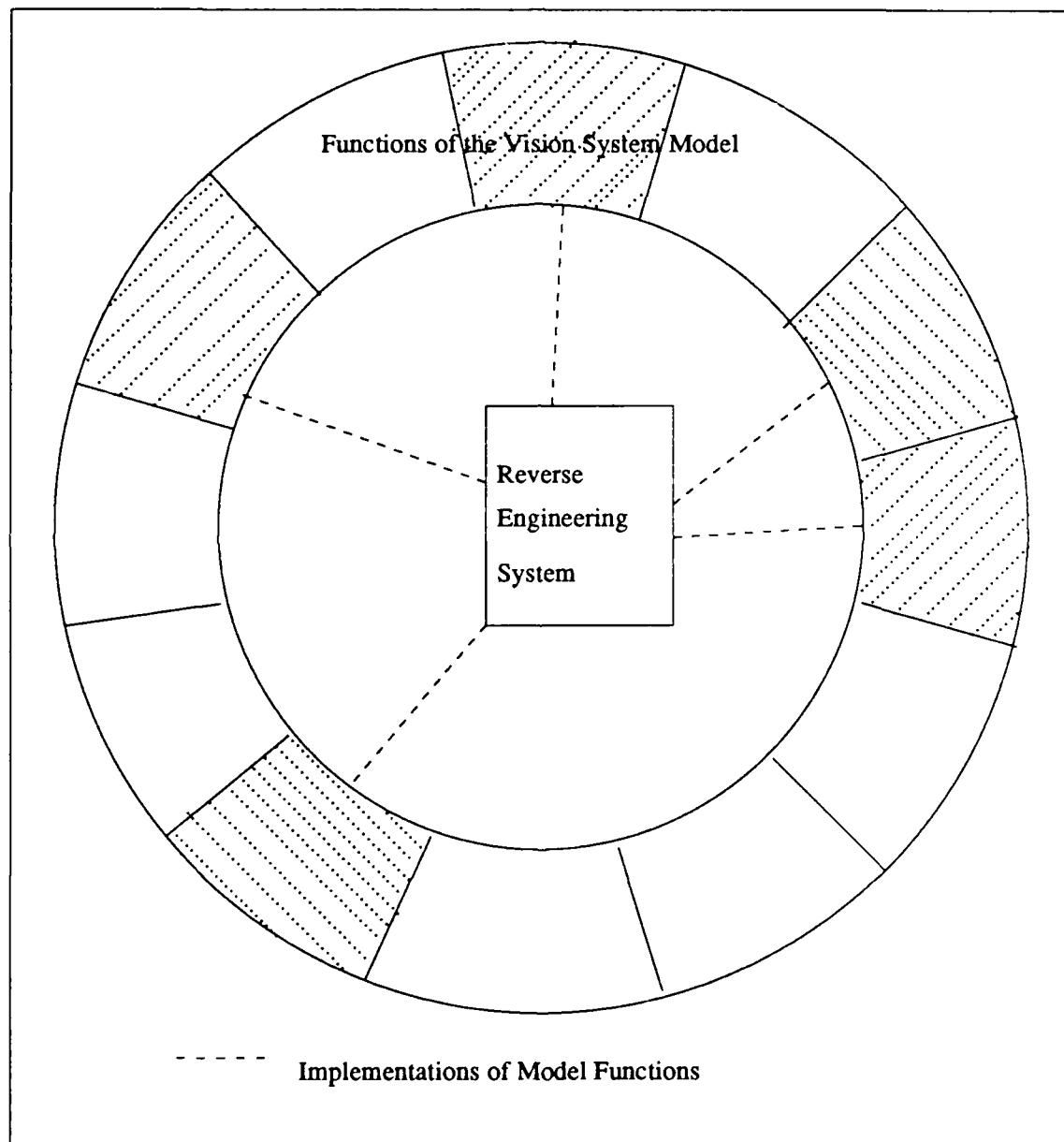


Figure 1: Relationship Between the Vision System Model and the System for Reverse-Engineering VLSI

The direction of the vision system model began by studying those elements required to produce a reverse engineering system. This system made a good test case with which to develop the model, because while the problem had a number of simplifying assumptions, it was still complex enough to require an extensive vision system. While developing the system, more effort was devoted to determining how the requirements fit into a general structure for vision systems than to constructing all of the particular portions of the reverse engineering system itself. In addition to the requirements derived from work on the reverse engineering system, explorations of how the vision processes function in animals and humans, both physiologically and psychologically, provided a basis for building the model. The final source of inputs into the construction of the model came from investigating technological approaches to portions of the vision problem. It is through this broad background that the model achieves its generality.

Kabrisky has proposed a simplification of the general problem of pattern recognition [Kabrisky 70]. He submitted that if we gathered a set of slides from a wide variety of sources, jumbled them together, and removed all color, when they are projected on a screen for a set time, we will still have an informative presentation. Despite having removed color, depth, time and sequence we will maintain most of the information in the set of slides. The same result is true in the real world; we could function well on only a subset of the information we receive. The recent history of our recording and display of visual events is sufficient to demonstrate this. For example, still pictures were not, and have not been rejected because of their inability to produce either ocular disparity or motion. Neither were early movies or television forsaken until such time as scientistists had developed color technologies. Indeed there is now a move afoot to keep color out of the older movies. Still, although we can function in our world less than the usual amount of visual information available to us, a blind man can function in the world even without that. So while it would be an accomplishment to build a vision system which can deal with shape, texture and grey scale without considering color, depth and time or motion, it is important to consider

what impact these would have on the function of the system. Therefore, any vision system model must have provision to eliminate, mitigate or incorporate these effects as appropriate to the information being extracted by the system.

1.2. Teleological Functionalism

As with any problem for which hints and answers are sought in an anthropological model, we must define our approach to whatever we may find. This dissertation will adopt the view of teleological functionalism [Sober]. This is a view in which objects and processes are defined by the roles they perform. In teleological functionalism, there is no isomorphism between function and process. The same function can be performed in many ways. There is also no limit to prevent a process from serving multiple functions [Sober]. Nor do we have a limitation which constrains a function to be performed in an isolable physical location. This view allows us to separate the form of a process from its function and adopt the one without the other. Without this we should be constrained to either, in the first case, accept the exact mechanisms of the brain as necessary [Turing machine functionalism - Sober; identity theorists - Place; eliminative materialism - P.M. Churchland; connectionism - P.S. Churchland and Sejnowski]. In this case we are left accepting the feathers of the bird as mandatory for flight, not as one type of mechanism which performs the function of creating a light-weight airfoil [Kabrisky 67]. In the other case, we accept that we cannot or need not understand the methods of the brain [behaviorism - Watson; Dualism - Descartes; Intentionalism - Dennett 90a; undiscovered physics - Penrose]. If we compare the plight of a boat builder to that of a scientist building a vision system, this second case is rather like claiming that there is no need to look at a car engine if we are trying to devise a boat motor for, either all that matters is that we pour in gasoline and sometimes a little oil to reward the car for taking us where we want to go, or the car motor does not make the car go - this is done by spirits who only are present for the warmth of the engine, or we cannot understand a car engine,

because all of the principles of physics have not yet been discovered. None of these views seem realistic.

By adopting teleological functionalism we recognize that in looking to the mammalian visual system, "as with any system of objects, the existence of function pretty much guarantees the existence of artifacts [Sober, p. 104]." We can functionally separate those features of the human mind which are needed for vision, even if those same features may also be used for speech, hearing, smell, or any of a number of other higher order functions of the brain. We also do not fall prey to the danger of basing our vision system too closely upon the human model. The danger in this is that people do not dwell on their shortcomings and failures as pattern recognizers and seers, but rather on their successes. This, along with a natural chauvinism, tends to influence our judgment of just how well our human vision system performs, and could cause us to require components which are needed only for a biological implementation of a vision system.

Our motivations for accepting a psychological approach to use in building our model are two-fold. First, if we are building a system to perform the same function as, or a similar function to, that of our visual system we cannot look only at the hardware in which it is performed. If we fully understand the function of every neuron and the way in which they are all connected, we do not have any more of a feel for the qualia¹ of the vision experience than an understanding of the transistors of a computer and their interconnections tells us about the function of a computer program. To fully realize the high-level functions which cause a vision system to perform in the manner in which it does, we must approach it from both ends: function and components.

Our other motivation for seeking a psychological approach is to help us comprehend when we have a vision system. We could build a vision system far more complex, capable and flexible than our own vision facilities, but if we cannot comprehend it in terms of a familiar psychological

¹The qualia define the vision experience as it appears to us internally. They are the color - redness, blueness, greenness - the appearance - fuzzy, crisp - etc. by which we quantify objects we view. For a complete description of what is involved in vision, as well as what is expected of a vision system, see Appendix A.

model we would not recognize it as such. Thomas Nagel has pointed out that we cannot truly know what it is like to be a bat as we have no comprehension of how its environment is built [Jackson]. However, we can project ourselves into a concept of what it is like to be a bat by mapping our visual psychology onto that of a bat. Similarly, we cannot know how or even if a computer can "see" and "recognize" things unless we can map these functions onto a familiar psychology.

Therefore, we look to teleological functionalism to provide us with a map to direct where to begin to look for, and how to look at, functions which need to be included in our vision system, and to tell us when we have accomplished that system. This approach will also allow us to explore a number of alternatives for accomplishing the processes required to assemble a vision system.

1.3. Problem Statement

This dissertation provides four major contributions to the field of vision research. The first contribution is a general vision system model. The model, which blends biological as well as technological methods into a coherent approach to vision, will provide a basis for implementing vision systems. The second contribution of this dissertation is to demonstrate particular implementations of portions of the model. These implementations will include methods for using Gabor wavelets in edge detection, in preprocessing images for use as feature vectors in backpropagation neural networks, and as basis functions in a recognition/reconstruction network, as well as methods for integrating color into a vision system. The third major contribution is an investigation of attention mechanisms using a two-part model with Gabor filters as a base attentional indicator. The final contribution is a description of an actual vision system for the reverse engineering of VLSI circuits in terms of the general vision system model.

1.4. Approach

Chapter 2 of this dissertation will present a general model for vision systems. The chapter will begin with an overview of the general model, and then will focus on particular portions in detail. Those portions will include the world view and world picture, the control system, the sensors and their transforms, the input transforms, and the super-conscious.

Chapter 3 will cover the implementation of particular portions of the general vision model. The chapter begins with a discussion of the Gabor wavelet and a detailed look at how it can be used to explain optical illusions in the human visual system. This is then followed by a number of sections which demonstrate how this tool can be adapted into implementations of the vision system model. These sections include discussions of Gabor wavelets for edge detection, for preprocessing images for use as feature vectors in a back-propagation neural network, as basis functions in a recognition/reconstruction network, and as the base attentional indicators in a two-part model of attention mechanisms. This model mainly covers the attentional indicators, but also provides some explanation of possible search strategies, the second part of the model. The final section of the third chapter will use thin film optics as a tool for building to a method of using color which will blend compactly into the vision system model.

In chapter 4, I will demonstrate the application of the model to the specific task of analyzing VLSI circuits. This will not involve an implementation of the complete model, but rather will be a demonstration of the types of considerations needed, and the means by which the vision system model can be implemented as a functional system. Finally, I will present several conclusions regarding the work performed and make recommendations for further study and consideration.

CHAPTER 2

The General Vision System Model

This chapter will discuss the general vision system model and introduce and explain the constituent pieces of the model. In the next chapter, I will further explore a number of constituent pieces and look at relevant biological/technical models which suggest means for the implementation of these components.

The General Vision System Model which I will provide is based on a combination of biological, technical and psychological models, as well as upon our perceptions of how we handle the vision task. The combination of models from these areas is important as the knowledge to allow us to build a workable model is not limited to any one of these areas, and because the combination allows us to build a more powerful model by taking the best elements from each area. This also fits with the teleological functionalist view we have taken in which we focus our attentions on the role each piece plays and its interactions with other players rather than worrying about the specific structure. At later stages this will allow us to choose the means which is most effective in the environment of a particular implementation. It is also important to understand that every implementation of the model need not include every functional capability of the model, but rather should include those relevant to the particular vision problem being addressed.

This model is only a preliminary attempt. It can be expected that as we learn more about psychology and how the mammalian visual system functions we will be able to improve this model and find better ways of adapting technological approaches to vision into the model. We can also expect a growth in the capabilities of these technological approaches which will further the capabilities of the General Vision System model. All of these improvements will rest upon the framework established herein.

2.1. Overview of the Model

A diagram of the Vision System Model is given in Figure 2. The central feature of the model is the world picture and the world view. The whole model is oriented toward constructing the world view. This view is constructed primarily by building a world picture which is then fitted into the appropriate position of the world view. The world picture is not unique; there may be more than one world picture under construction at any given time. The Vision system may have one or several outputs, all or none of which may be active at any given time. The outputs of

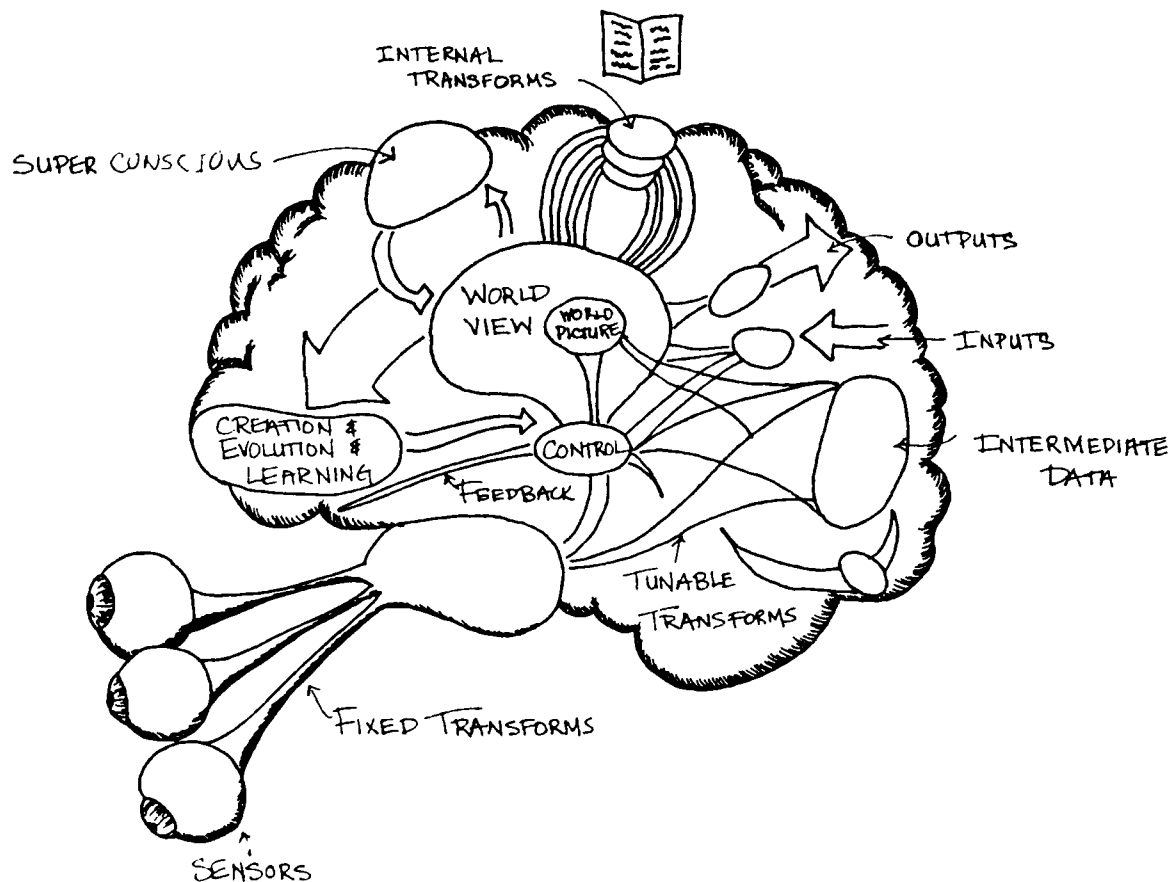


Figure 2: The Vision System Model

the vision system are either the world view or a set of features filtered from the world view. The model has a set of operations which perform this filtering. All other portions of the model are oriented toward creating, improving, or deleting portions of the world view and world picture. The employment of the operations acting on the world view and picture is orchestrated by a control section which makes decisions based on observations of the world view and picture. This control system somewhat imitates Homunculus, the little man inside of our heads who controls what we are thinking, although in this case his actions are restricted to those dealing with vision.

On the left side of our diagram is a series of input sensors. These sensors may possess a number of characteristics which can be controlled by the vision system. Such characteristics may include focus, position and other factors. For each of the sensors there is a set of characteristic transform processes which cannot be affected by feedback from the vision system controller, although they may obtain some degree of feedback from within the immediate sensor process. These processes bring the input to an intermediate state where it can be further processed to provide data to develop the world picture or to allow the control system to provide feedback control to the sensor from which the data were obtained or to control other sensors. From the intermediate stage there are a number of parallel processes which transform the data into pieces which compose the world picture. In general, all sensors contribute to the construction of one world picture, although there may be cases in which more than one world picture is under construction simultaneously. In these cases each world picture is constructed from the inputs of unique sets of sensors.

Immediately above the world picture is a series of operators which use as their input the world picture or the world view. Some of these may be specifically oriented toward operating on the data contained in the world picture and make their contributions only to the same. Others may either accept as their input data from anywhere in the world view, and provide output to a specific world picture, which may at times not be the major world picture under construction.

These operators may run constantly, on a predetermined schedule, or on the request of the control system. Details of their operation may also be affected by feedback from the control system. These operators be any of a number of different types of systems to include correlators, expert systems, model-based recognition systems and others. They may maintain their own secondary knowledge and databases, but these information stores are composed of information to help their processes, or of intermediate results. By their nature they are ancillary to the vision system and do not provide results external to the system, although in practical situations they may be observed in order to discern information about the condition of an implementation of a vision system being constructed. It is also possible that the desired results of a particular output filter may match the supplementary store of an internal operator. In a practical implementation these may be created by the same process, but for the theoretical model they should be considered as separate entities.

A significant constituent operator is the intuitional operator or "super-conscious". This operator is shown to the right of the world view and may consist of one or more semi-independent units. These units watch the operations on, and the development of, the world picture. Their function is to provide an intuitive type of capability to the vision system. I will refer to these units collectively as the "super-conscious". The super-conscious attempts to make tentative identifications of items in the world picture and to paint in gaps in the world view. The product of the super-conscious is not considered to provide an exact answer, but rather to provide the types of functions that an intuitive sense provides in an individual by providing suggestions for processing methods, models for model-based reasoning and goals for theorem proving.

Finally, at the bottom of the model we see an arrow labeled "Creation & Evolution & Learning". This area represents the realization of the model into a functioning system, and the tuning of that entity to the specific problem, or set of problems, which the system handles. The creation of the system is done externally; however the evolution may either be done externally, or

internally in response to some changing external conditions. In general there are two types of evolution to be considered. The first type of evolution is used for general improvements and adaptations of the vision system. Another more specific evolution, or tuning process, may be performed in preparation for accomplishing a specific vision task, after which the system will return to its original state or await tuning for its next task. Some vision systems may retain knowledge about the way in which they were tuned for a particular task in order to apply that knowledge to later tasks.

2.2. The Model's World Picture and World View

The world picture and world view are the central features of the vision model. All processing is focused on either contributing to their creation, or filtering their contents for output. The world view is a large pictorially oriented database. The world picture is a particular region within this database on which the major attention of the system is focused at any particular instant in time. Processing is not limited to data contained in the world picture, but the major input functions should all be cooperating to create a world picture. This world picture will then be fitted into its proper place in the world view.

Concentrating the sensors and their transforms/interpreters on the world picture at one time allows a number of advantages. First, a degree of synchrony in both orientation and time is obtained. This eases the problem of combining multiple sensors. If the sensors are allowed to operate independently, it introduces additional registration problems which need to be resolved for each set of sensor windows. If, however, the sensors operate in synchrony, the number of registration calculations needed is reduced. This is true whether the sizes of the fields of view of the sensors coincide or not. Further, the size of the remaining registration problems tends to be reduced. This happens because after an initial registration has been calculated for a particular world picture, the remaining calculations are small adjustments from predetermined offsets.

Time-sequencing helps to eliminate conflicts created by changes in the input environment. Objects in the environment will maintain the same relative positioning and other relationships, which if observed at different times might vary and cause confusion among the results from the sensors. Time-coordinated concentration of sensors also allows the vision system to gain the maximum amount of information about one particular world picture, as often the whole of the results may be greater than the sum of sensors' capabilities.

Concentrating the sensors also provides a simpler sequencing for the control structure and lessens the amount of data needed to be stored. If the sensors are not operating in concert, a great deal of extraneous data needs to be retained either for sensor registration, or to maintain separate input transformations and interpretations until such a time as they can be properly combined. Coordinating inputs allows the outputs of the sensors and their transforms and interpretations to immediately support or deny the results obtained from other sensors. Thus the storage requirements are reduced, and at the same time the probability of the system's maintaining, for some period of time, an erroneous interpretation is reduced. Finally, operations which work on the contents of the world view can be assured that all available data are in place and no new inputs will be added without first switching attention back to the area in question. This helps to ensure data integrity and prevents operations from taking unnecessary multiple looks at the same portion of the world view.

The world picture may be required to have a much greater detail and richness to its contents than may the world view. The purpose of this added information is to aid in creating the coherent scene in its proper perspective within the world view. The added information may also be needed for some particular set of processes operating on the world picture. This extra information can be thrown away when it is no longer relevant, or it can be stored away in a long-term location for recall when attention is again focused on the particular world picture from which it comes, or on some similar alternative.

The current location of the world picture within the world view is controlled by an attention mechanism which considers external directions, the location of the sensors, the requirements of the internal processes, intermediate results of the sensor transforms, and the state of the world view. Each of these must be considered in relation to the other elements.

The world-view visual base must be capable of encompassing a complete description of objects needed for the required outputs of the system and any environmental modifiers needed for their output. These must be incorporated into a framework with the flexibility to provide and maintain appropriate relationships among objects and their modifiers. A sample world view might be that needed for a tank recognizer. The world view must be able to represent the type of tank, its friend or foe status, its orientation and other significant data. It must also be able to represent sufficient information about the terrain in which the tank is operating to allow the system to maintain the tank's relationships with masking terrain, such as hills and trees, and obstacles, such as water, ditches, etc. The world view must be able to place all of these in a framework which includes distance, azimuth, and elevation information.

In the human brain the problem is even more complicated because of the potentially unlimited number of objects and relationships which must be maintained. In addition the framework in which these objects need to be fitted is not necessarily the simple three-dimensional world that appears at first glance. Indeed, we must consider the case of a theoretician who is contemplating an equation. As he sits in his office at his desk, he has a world view which encompasses the contents of that office. Initially, his world picture may be of the paper spread out before him on his desk, but then he begins to draw in his mind a set of equations. The place in which he draws these is not in any of the three dimensions of the world of his office, and yet it is there drawn on a hyper-plane of his world view. Likewise, he may begin yet another hyper-plane upon which he draws three-dimensional representations of the meanings of the equations. Then to confuse us even more, he flips his world picture back and forth, from the drawings to the equations, like the

pages of a book. Still other times he shifts his world picture back to the paper on the desk in a seamless motion.

Not all world views are as complex and demanding as that of the human brain. In a simple bar code reader, the entire world is made up of the bits which represent codes and a symbol for a non-code or read error. Other systems lie along a continuum somewhere between the ultra-simple world of the bar code reader and the ultra-complex world of the human mind. Each system has the capability to represent and contain those events which are significant for its particular task.

The representation encapsulated in the world view is not exclusively a final representation. It can include rough observations which will be refined to a better representation as greater accuracy in the recognition process is needed, or as more information about the area or objects becomes available. Thus, what may start out as an entry of a brown moving patch may become a dog as the recognition system is given a description of what a dog looks like and as it refines its observations to find four legs and a tail. Possible alternatives to a final depiction may include: significant descriptions captured, but insufficient to complete a recognition; erroneous recognitions awaiting discovery and removal; and assumptions and inferences or projections presented as possible solutions to a particular recognition problem. These projections may include predicted movement of objects, the occluded portions of objects, or inferred objects which have not been included in the field of view.

Given a world view in which a room is being constructed, it would be entirely reasonable for the recognition system to construct a back wall to the room even though it is not in view of the sensors and has not been filtered through the world picture. This inferred wall would be constructed in a position geometrically consistent with the other known walls of the room, and would be given an appearance consistent with that found on the other walls. If the center wall of the known three walls appeared to be rather distant and the walls gave clues which indicated a small

room, the inferred wall might be placed rather close to the position of the vision system. On the other hand, if the center wall were somewhat closer the inferred wall would be placed farther away in an attempt to maintain a consistent geometry. The style of the inferred wall could be brick to be consistent with the appearance of the other three walls. These illusions in the world view are maintained until other perceptions alter them and allow new inferences, or the wall is viewed by a sensor, constructed in the world picture, and updated in the world view. At times this update may be quite extensive and result in a flood of updates of other inferences which were based on the particular model used for the wall in question. Imagine a room with three walls of grey stone and tiny windows where the center wall is fairly small, and the fourth wall is a wood-framed picture window a good distance away with a view of a forested lake and mountains. The perceived picture changes rapidly from a prison cell to vacation cabin. In this manner the world view progresses from rough observations and projected features into a refined representation of the recognition world.

2.3. The Control System

The control system is responsible for coordinating and directing the activities which combine to create the world picture and view, for determining the focus of the world picture, for controlling the makeup and contents of the world view, for selecting and directing the external sensors and for maintaining the goals and current state of the vision system. It also provides communications with the external environment. To accomplish these tasks the control system uses a number of attention mechanisms, which are feed-forward paths from the system inputs, the world view and picture, operations on the world view, and the external world. It also provides feedback to tune the sensors, the input transforms, the output transforms, and the world view's operators. In order to be able to accomplish these tasks the control system must monitor the current state of the system and be able to suspend or resume any particular state.

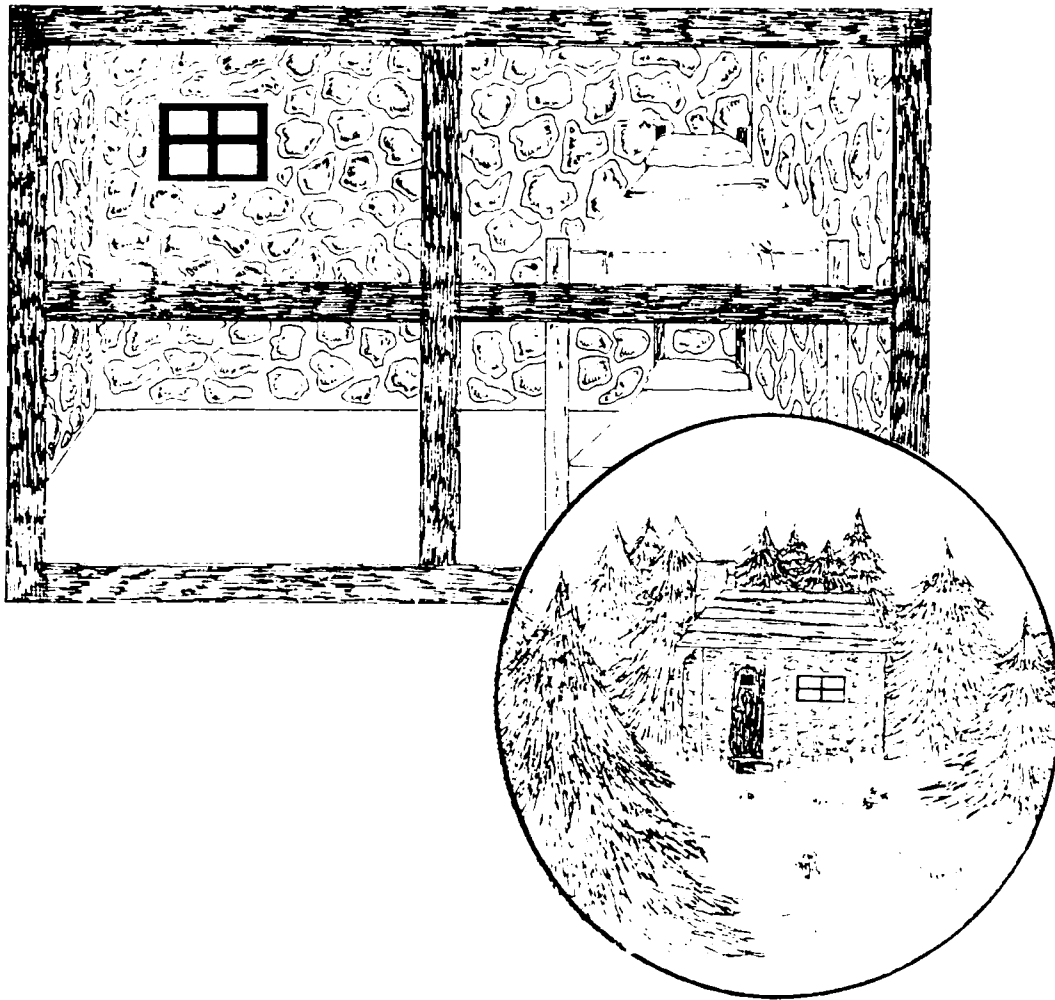


Figure 3: A Deceptive Close Up

2.3.1. Maintaining Goals

The control system maintains and directs vision system activities toward accomplishing some set of vision system goals. These may be as simple as the hard-wired goal of a bar code reader to interpret bar codes, or they may consist of a much larger mutable set. Some goals are directly oriented toward producing the outputs of the vision system, while others may be more concerned with accomplishing internal tasks which facilitate the larger goals.

Goals which determine the system outputs may include such things as: interpret bar codes; find all airplanes; find specific types of airplanes; or identify defective parts. These may be further refined by other goals which modify the system's objectives to accomplish a particular

task. These goals may include such things as: locate a particular bar code; find airplanes in this area; identify F-16's and F-14's; or find all faulty widgets. These types of goals are usually externally inserted into the system; however, in some cases they may be the result of an evolutionary type of action.

Goals which help a system to accomplish its primary output goals are usually not acquired externally except during the creation of a new system. Instead, these goals are developed by internal processes, or the set of goals inserted into the system at its creation is modified by evolution. Simpler pattern recognition systems may rely almost entirely on the coding of goals during creation. However, systems dealing with more complex problems must rely more on the internal creation of intermediate goals. This goal creation may be guided externally, but if the vision system is sufficiently complex, and the problems it is asked to perform sufficiently difficult, the system creator cannot expect to have anticipated all of the internal goals necessary for performing the vision tasks, nor is it reasonable to expect that anyone will be available who understands the internal workings of the system well enough to construct reasonable internal goals. The problem of anticipating internal goals becomes an even more dominating factor for a system in which the environment or the target do not remain stable.

2.3.2. Attention Mechanisms

Attention mechanisms are important for establishing the location of the world picture within the world view, developing internal intermediate processing goals, focusing the world picture on specific regions or problems of the external world, and for constraining the model to maintain focus for sufficient time to perform useful processing. Attention mechanisms are actuated by any one, or a combination, of four methods. In the most direct method the attention mechanism watches the incoming data from the sensors. It may need to have a transformation applied to the data to provide attention cues. In the optimal case, this transform would be pig-

gybacked off of a transform needed for building the world picture which either matches in part or in whole the requirements of the attention mechanism. The attention mechanism may also be stimulated by certain events or combinations of events filtered from the world view itself. Stimulation may come from a process which finds that it is lacking certain information in a particular area, or that the presence of some feature is indicated for a particular region. Contraindications could also prove useful to the attention decision. Finally, external stimulations may affect the attention mechanism.

The control system must combine and sort the incoming stimulations and use them to make a decision about where attention should next be focused, and about how long the attention is required/desired. It must use some type of priority system to decide how to distribute attention when there are two or more areas which require study. Within a particular scene or world picture the control system must use its attention mechanisms to determine which events are significant and need to be further defined. It will establish whatever new goals are necessary to focus the system on making these determinations and will decide which functions are most appropriate for pursuing these new goals. The control system must also have a method to govern when one area competing for attention will interrupt ongoing processing in another region. The control system must also decide which portions of the vision system are appropriate for the needy region. All of this must be accomplished within the context of the goals of the system.

The control system must also decide where to focus the attention of the system when there are no stimuli competing for attention. The idle search strategy the vision system uses is important as it will determine which attention mechanisms are able to be aroused. To be most successful the vision system must be able to anticipate where the next attention spot will be. For simple systems, such as a bar code reader, this is not a problem. The idle focus is at the end of its wand. But, for more complex systems the idle search must be much more carefully designed.

2.3.3. Long Term Storage

The control system is also responsible for determining what types of information get placed into long term storage for the vision system. There are three types of candidates for long term storage. They include state information for context switches, significant events, and the routine storage of predetermined lists of data. The storage of state information is used to allow the control system to shift the focus of the vision system at times determined by the attention function. This storage mechanism usually involves large sets of data which will be accessed as a group and which generally need not be maintained after the group has been restored. Access to the data within a state is usually ordered, although access to the particular state may be random.

Significant events are unique one-time or recurring circumstances which intrinsically excite some type of attention function to cause their storage. The significant events are not stored as a routine matter, but only as the result of some circumstance which either matches an entry on a significant event list, or which in and of itself generates a flag for a significant event. These may include such things as unusual arrangements of scene components, novel scene features, or other abnormal phenomena. A completed world view or picture may also be seen as a significant event in some cases, but may not be viewed as such if it were completed on a routine schedule. Significant events are generally considered to have future significance to the operation or results of the vision system. Data stored for significant events are generally less voluminous, but are also generally accessed as a group although the data usually will be maintained after access. Access to these data is usually random.

Routine long term data storage is accomplished to aid the vision system by reducing the amount of information the system keeps in its immediate storage, to secure information for future use, and to maintain lists for which there is no immediate requirement. Typical of the types of items placed in long term routine storage are: labels or names associated with objects; portions of the world view not of immediate interest; lists of objects found in a given reference frame and

completed object descriptions. Such items are usually stored as the result of a time schedule or as a result of an item's appearing on a save list. These items are not considered to be significant in and of themselves, but the long term maintenance of these items will contribute to the success of the vision system. The individual items stored are not usually large, but they can be expected to be individually accessed. Access to these data can be expected to be both random and ordered. The items must usually be maintained after access, but many items will be stored in shortened queues, or structures which function similarly to queues, where old data are destroyed or consolidated as new data are stored. Much of the data stored will also have time value and thus will be subject to erasure, deletion or overwriting after certain time intervals. These intervals may be predetermined, or may be based on the volume of information maintained in storage, or on some combination of the two methods.

2.3.4. Feedback

Inherent in the attention mechanism, the focus of the world view and picture and the development and maintenance of goals has been the necessity of feedback to the sensors, the input transforms and the processes working on the world view. Feedback is also important for tuning transforms, processes and sensors. Although they may have localized feedback to provide some degree of control over their function, sensors, processes and transforms can only be assured of being useful to the system by feedback from a level with knowledge of the use of the results of that element. For example, a filter may be used to trim off all but the highest 30% of the amplitude of a signal coming into the system. It will be able to analyze its output and adapt to changing signals to maintain the required 30% in all circumstances, but yet there may be times when receiving only the top 25% of the signal provides a better result to the overall system. It is this type of high level tuning that the control system is expected to perform or, if it is unable to, to ask to have done through external intervention.

2.4. The Sensors and Their Transforms

An implementation of the vision system model will have one or more input sensors. It is the responsibility of the sensors to capture some particular property of the external environment in a form which can be used by the vision system. One method which can be used is to have a variety of sensors, each tuned to some particular set of properties. Another method would be to use multiple sensors with identical properties, each of which is responsible for some area, which may overlap the areas of other sensors. A third option would be to use a set of sensors which traded area coverage for accuracy. With this set the broad-area, courser-accuracy sensors would be used first for attention focusing and macro events. As events become isolated and more information is needed the finer-detail sensors would be used at the loss of some broader area of coverage. These three methods could also be used in any number of different combinations.

The mammalian visual system tends to use an interesting combination of these methods. Each eye is made up of two different types of photoreceptors. These each have their own separate sets of characteristics. The rod cells are more sensitive in low light situations, but their inputs in the central foveal region become less meaningful when compared to those of the cone cells as the intensity of the light increases. The cone cells are less sensitive in general, but are finer tuned for frequencies to which they will respond. As the intensity of light decreases, the sensitivity of the cone system rapidly degrades until a point is reached where the rod cells become predominant. This transition can be seen in the dark-adaptation curves of Figure 4. The increase in sensitivity over time slows as the cone cells reach their full potential. At this point the sensitivity curve flattens somewhat. As the rod cells begin to adapt, the sensitivity again increases until the rod cells reach their full potential. The eye maintains a higher level of sensitivity than would be obtained if only the cone cells were used. The mechanism to accomplish this appears to be a pathway through the amacrine cells which becomes effective under conditions of dark adaptation while more direct links to the ganglion cells shut down [Sterling].

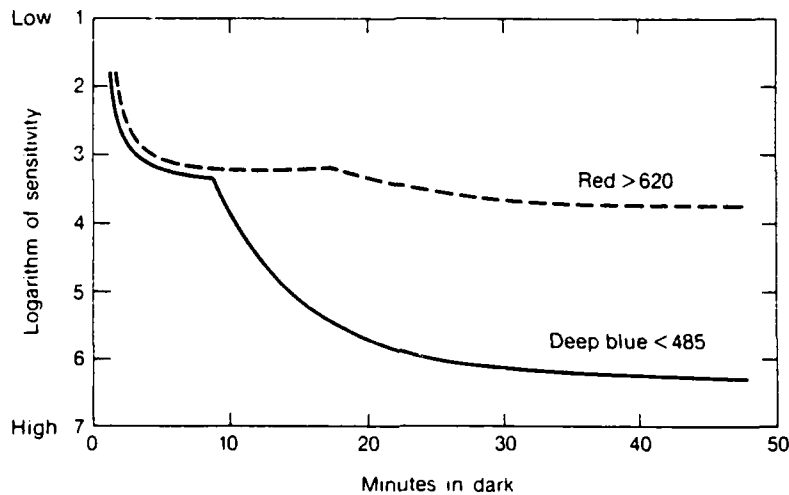


Figure 4: Adaptation Curves for the Eye Under Varied Lighting Conditions [Goldstein, p. 81]

Within each eye the placement of the photo receptors is arranged with the highest density of cone cells in the central foveal region. Within this region the highest possible degree of information is gathered from a scene. The dense concentration of the cones not only allows images in the foveal region to be encoded as a three-dimensional color space, but it also allows the image to be encoded in great detail. Further from the fovea, the rod cells become more dominant and, in fact, the concentration of all types of cells falls off. This results in a larger visual field with a lower degree of accuracy and a simple greylevel color map. This map is shifted to a shorter wave-length portion of the spectrum, where more energy will be concentrated during periods of lesser illumination.

The eyes, each with its complete complement of receptors, are then duplicated and given overlapping fields. This allows for the use of binocular disparity as a measure of distance. In some other animals two eyes are used not just to measure distances, but also to increase the size

of the overall visual field; this is achieved by setting the axes of the eyes at an angle greater than 180 degrees.

A vision system can take advantage of all of these types of sensor employment and also can take advantage of others because it does not have the biological constraints which are placed on natural systems. For example, the sensors do not need to be placed in close proximity to one another in order to be able to communicate, as current communications systems are much faster than nervous systems. Nor is a vision system limited by the body configuration of a mammalian host.

The outputs of all sensor transforms do not need to add specific events to the world picture. Some transform results may be used instead to adjust parameters of the world picture and world view. They may perform such tasks as determining the scale in which the results of other transforms will be entered into the world picture. They may provide information on the relative extent of the world picture, or they may provide some sort of global modification to the world picture or to how data is processed into it. An example of such a modification might be a detector to determine whether or not it was dark or nighttime in the external environment. Knowing the ambient light condition would both provide guidance for adjusting the other external sensors and their transforms, and allow the world picture to be adjusted to use a different interpretation of the information obtained from the sensor transforms.

The Teleostei, a subclass of vertebrate fishes, possess such a regulatory organ in the form of their pineal organ. This organ has photosensitive cells, but relays no image, edge, or movement information. The purpose of the organ appears to be to provide coordination for the circadian cycles of the fish. Ekström and Meissl studied the pineal organ of rainbow trout, *Salmo gairdneri*. They found that the organ not only produced afferent signals with a spontaneous firing frequency inversely related to the intensity of the light, but that it also received efferent projections from other brain areas. The studies indicated that the signals from the pineal organ are modulated

by a number of influences. The researchers felt that it was important that the organ's output be modified to allow it to be insensitive to irrelevant stimulations and to adapt to periods of adjusted metabolism, such as during the reproductive period [Ekström and Meissl].

2.5. Input transforms

The input transforms of the vision-system model are those which perform functions on the sensor data prior to their being inserted into the world picture, but which can be adjusted by the control or other selected sections of the vision system. The sensors themselves are controlled by the model in so far as they can be directed at particular targets, or zoomed and focused, but the manipulation-changes to the outputs of the sensors are, in all normal circumstances, driven by the inputs to, or internal states of the sensor itself. Thus, the input transforms represent a unique stage in the vision system model in that their interpretation of the sensor data to fit the world picture is adjustable from the state of the world picture itself. These transforms may also have some capability for adjusting their own performance, but this is subservient to control from the overall system. Thus, while the ability to adjust the input sensors serves as a recognition of the noisiness of the external world, the adjustability of the input transform recognizes both the noise and the ambiguity of the external world and provides a mechanism for interpreting that world in proper context. In general there are two types of input transforms: those that place data directly into the world picture, and those which are used to establish the parameters or environment for the world picture.

2.6. The Super-Conscious

The "super-conscious" is unique among the operators working on the information in the world view, in that it is not attempting to directly affect the contents of the world picture. Instead the super-conscious uses the information it finds to influence other processes and to make conjec-

tures about the nature of the information stored in the world view. It does this by extrapolating from the information which is contained there and trying to assimilate it into some more detailed or broader view. "To understand what an individual fact comes to, the system will have to place it within a larger organized structure of facts (some analog of a theory) [Van Gulick, p. 114]." The super-conscious provides then a link between the low-level possession of bare facts and the higher structure needed to allow the system to use these facts to build the desired representation of the world about it. This is accomplished through the creation of goals, which can either be as formal as the goals of a theorem proving system, or as simple as a request to the sensors and their transforms for more information about a particular area. While there is no structure in the "meat" which directly represents this "goal stack", the function of the brain gives the appearance of operating in a goal directed manner. These goals can be represented as desires or beliefs of the vision system. In satisfying these desires or testing these beliefs the vision system furthers the development of its world view. The facilities of the super-conscious are also essential if the vision system is to be self-adapting. "Just as possessing information presupposes the having of goals, so also no system could adapt its behavior in the ways required by our analysis of goal directedness without *ipso facto* possessing information [Van Gulick, p. 113]." Self-adaptation, goals, desires and beliefs are necessary parts of a human-like visual system.

The super-conscious is the portion of the vision system which when presented with three walls, fills in the fourth to make a room. It is the portion of the vision system which when given a brown moving mass poses the possibility that the mass could represent a dog. The outputs of the super-conscious are not taken as facts or proven theorems, but rather as conjectures, possibilities and options from which search strategies, goals and working hypotheses are formed. The super-conscious should be formed with the greatest degree of flexibility possible and should have an inclusive approach. That is it should take the broadest approach possible from the available data, and work to exclude only those interpretations of the world picture which are excluded by

the most cautious interpretation of the facts. This is not to say that it should present all of its suppositions at one time, indeed they should be carefully filtered, but rather that it should not exclude any possibility however unlikely.

As the objective of the super-conscious is to focus the search for items in the recognition space, the approach should be to choose the most likely alternative solution possible given the available information. Once the solution has been proposed, the control mechanism of the vision system can direct the efforts of the sensors, the input transforms and the inner processes to either confirm or deny the proposed solution. With a system like this it can be expected that it will be easier to disprove than to prove many assertions of the super-conscious. For this reason, the vision system must also contain a mechanism which encourages it to also explore alternative possibilities. For those cases in which the vision system cannot confirm, or can only partially confirm, the proposed solution of the super-conscious, the system must encourage further proposals. The mechanism of the super-conscious must allow for this.

The requirement to be able to both provide multiple suppositions and to be able to propose a solution based on even extremely limited data means that the super-conscious must be able to draw its conclusions from increasingly complex abstractions. This is similar to the elephant-in-the clouds capability found in people. A group of young boys, lying on their backs on the hillside staring up at the clouds and watching, begin to search for shapes in the clouds. The first of these boys, being very practical says, "There are nothing but clouds up there." The second, however, replies, "No, I see an elephant over there."

"Where?"

"Well, if you look at that big grey cloud just over the tree, he has his face on the left. That little cloud is his trunk spraying water, and that wisp way over to the right is his tail."

"Oh, now I see it."

This scenario shows two differing cases of increasing abstraction. The second boy lowered his requirements for the degree of fit to a whole range of objects in his store of knowledge. He may not have lowered the fit for everything as we can expect that he would not have found a math book in the cloud formations. But it would have been entirely reasonable for him to have spotted a camel, a pirate ship or a firetruck. Once he did find something, he was able to focus his increased abstraction to locate the pieces of the elephant. The first boy used a more specific type of abstraction. He allowed his requirements to first be lowered for the specific parts of the elephant and then fit these into his elephant model. At this point the reader has also joined in the abstraction and has placed an elephant facing left, with a grey billowing body, into his world picture.

This model shows the super-conscious as the generate section of the "generate and test" paradigm commonly used in AI. This is a model which Dennett claimed "is a necessary feature of all modes of learning, and hence a necessary principle in any adequate psychological theory [Dennett 90c, p. 71]." We can extrapolate from that and infer that any sufficiently general vision model must have some learning mechanism, and therefore, that it must contain some generate and test mechanism. This becomes even more clear if we pose the task of the vision system as learning the contents of the external world. In this context, the super-conscious becomes the global generation mechanism for the vision system. Other portions of the system serve as the testers for the system. This does not preclude the inclusion of generators within the subportions of the system, nor does it require that the super-conscious be the generator of the final solution to a particular task. Rather, it provides a generation mechanism which is not tied to a particular process.

Why then do we call this the super-conscious? In general, we are not conscious of the generation of possible solutions. The "ah-hah" phenomenon of receiving a sudden inspiration is familiar to all, as is the case of a solution to a problem coming while we sleep or have in other ways cleared the problem from our conscious minds. Indeed, mindlessness seems at times to be a

necessary condition for the generation of solutions to many problems. Yet, we are often cognizant of the selection process. Once we have been presented with a solution by our generation process we go about mentally testing it to see if it is indeed valid. The generate and test mechanisms are not always split along the conscious/unconscious boundary, but the division often appears to lie near it. A composer with this alignment was Mozart: "When I feel well and in a good humor, or when I am taking a drive or walking after a good meal, or in the night when I cannot sleep, thoughts crowd into my mind as easily as you would wish. Whence and how do they come? I do not know and I have nothing to do with it. Those which please me I keep in my head and hum them; at least others have told me I do so [Dennett 90c, p. 75]."

"The inferences we attribute to rational creatures will be mirrored by physical, causal process in the hardware; the *logical* form of the propositions believed will be copied in the *structural* form of the states in correspondence with them [Dennett 90b, p.164]." This then is our Super-conscious.

CHAPTER 3

Mechanisms for a Vision System

This chapter will explore a number of mechanisms which can be used in implementing a vision system. Many of these mechanisms have their origins rooted in biological models. This should not though suggest that these are the only or even the "proper" mechanisms to be used for the purposes which they serve. Neither should they be avoided for their origins. Instead they should be applied where they match the requirements.

3.1. Wavelet Models

The study of wavelets has received a great amount of attention in recent literature in a number of fields. Some of this attention has been due to dissatisfaction for certain applications with other analytical techniques, such as the Fourier transform. Some researchers have been attracted to wavelets because of intrinsic mathematical properties of wavelet representations. Others are attracted because of the closeness with which wavelets approximate phenomena they have observed. The concept of the wavelet is a simple one. A wavelet is a function confined to a localized region. Wavelets can also be defined in more rigorous mathematical terms, but for our purposes, the simpler, more general definition is sufficient.

Wavelets can be described as having two components. The first, a modulation function, is enclosed in the second, some type of limiting envelope. The modulation function may be some type of periodic function, such as a sine wave, a square wave, etc., but this is not always the case. It is possible to use a step function or some other non- periodic function. The envelope of the wavelet can take the form of a spatially limited Gaussian Window, a limited exponential function, a single period of a square function, or some other spatially limited function (Figure 5).

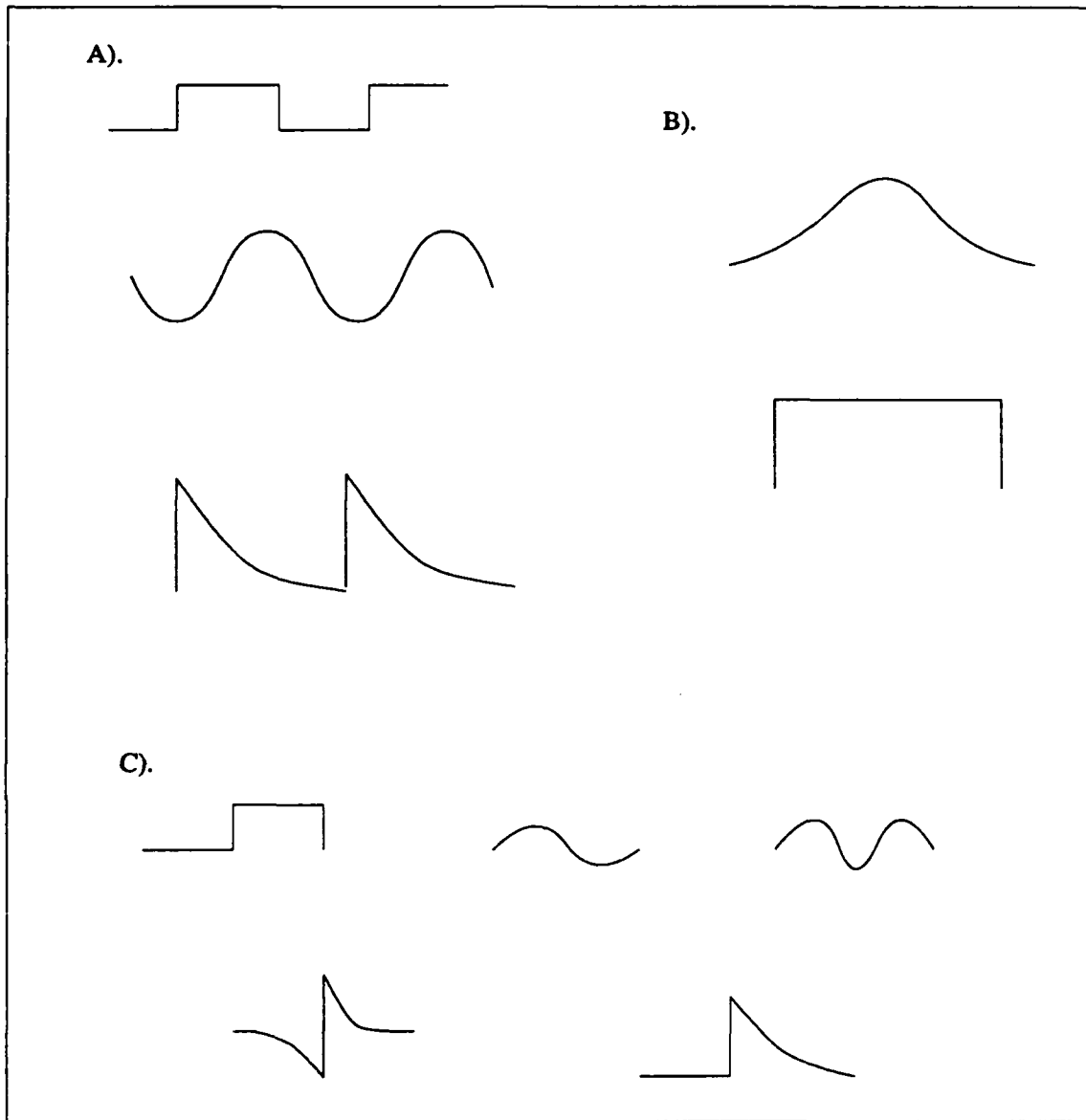


Figure 5: Wavelet Samples: A) Modulation functions
B) Envelopes C) Wavelets

Wavelets are not limited to single-dimensional functions, but have also been described in two- and three-dimensional implementations. Each of these wavelets is localized within its reference system.

In addition to changing the types of functions used for the wavelet components, the parameters of the function themselves can be changed. Wavelets can be described with specific spatial

orientations. They can contain a varied number of periods of the modulation function. The modulation function can be altered in phase; that is, its location is shifted in relation to the location of the window. The size, and the shape of the envelope can be changed in one way or another. If the number of periods of the modulation function contained within the envelope is held constant and the size of the envelope is varied the resulting set of wavelets is said to be self-similar. If the sizes of a set of self-similar wavelets are varied by some regularity (linearly, logarithmically, etc.) the wavelets are then said to be an "affine" set (Figure 6). Wavelets can also be grouped into orthogonal sets. In this case the set contains wavelets which possess differing

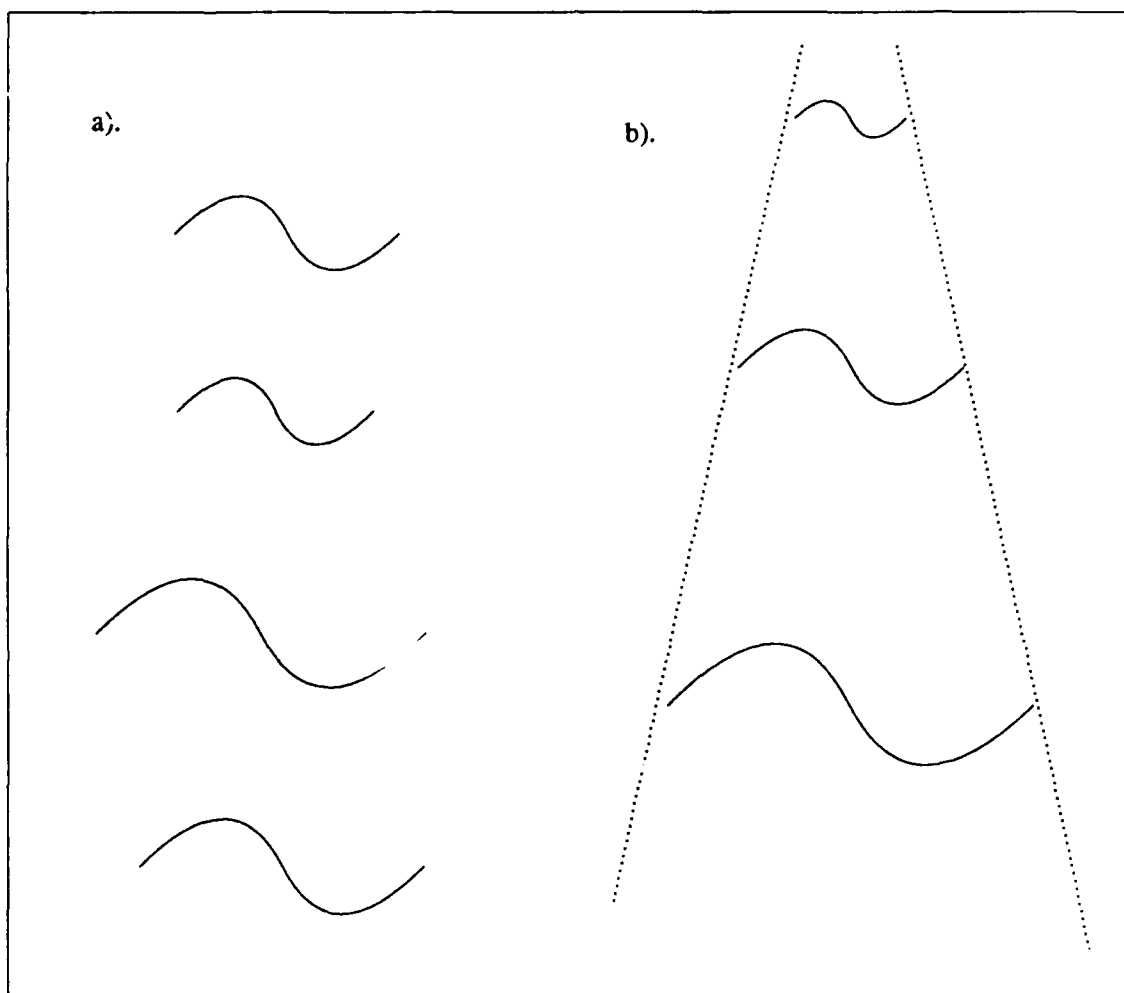


Figure 6: Self-Similar (a) and Affine (b) Wavelets

orientations and sizes such that the spaces they cover on a frequency plane minimally overlap. The annulus described by such a set of wavelets in Figure 7 would be duplicated by subsequent sets of wavelets with the same orientations and a smaller size. Orthogonal sets are affine.

Orthogonality is an important construct for mathematicians for it allows the complete space of solutions to be covered without redundancy. The concept has its place in the coding and reconstruction of images for if we wish to make a more compact code, orthogonality reduces the repetition within the code. Spanning the feature space is also important for lossless coding, but for recognition and other vision mechanisms, spanning the space and reducing the size is not as important as insuring that the important features are well covered. As a result, for vision systems, the requirement for orthogonality in wavelets can generally be relaxed.

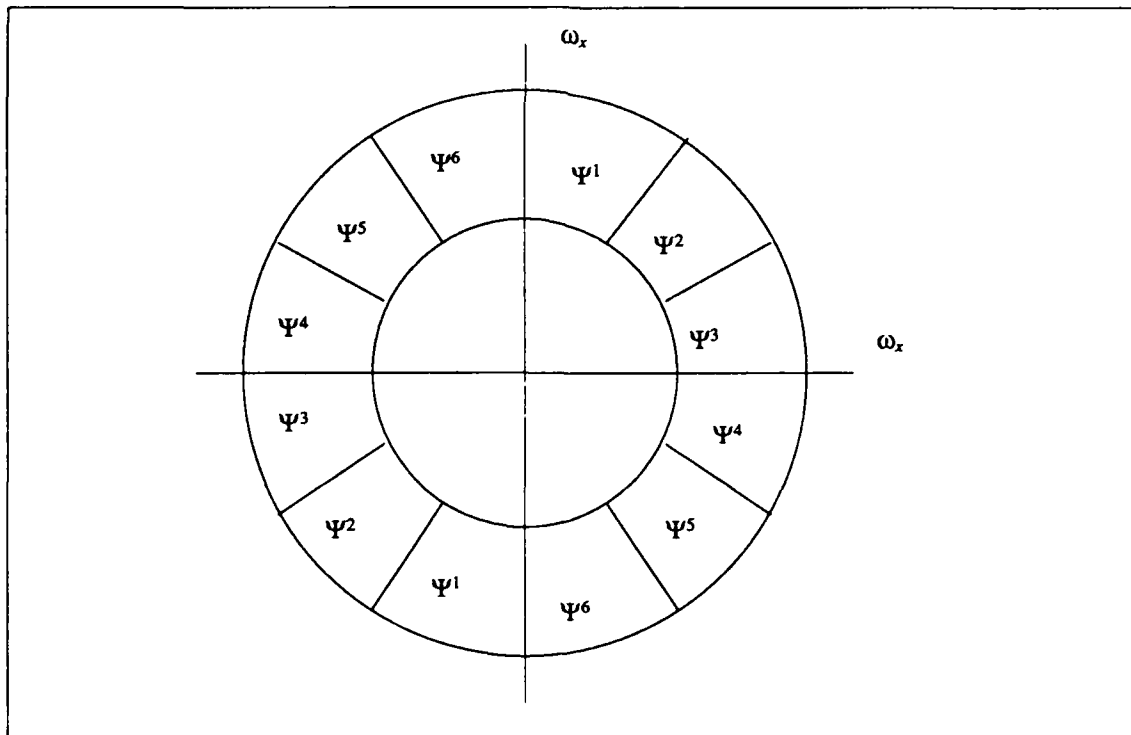


Figure 7: Annulus of Coverage for a Set of Equal-Sized Wavelets [Mallat 89a, p. 35]

3.1.1. Gabor Wavelets

One set of orthogonal wavelets is the Morlet wavelet. In this wavelet a sinusoid is contained in a spatially limited Gaussian wavelet. The envelope size is generally one or two times the period of the sinusoid. Envelope sizes are increased exponentially. A generalization of the Morlet wavelet is the Gabor (or Cubic Spline) wavelet. The Gabor wavelet drops the requirement for orthogonality. The equation for a Gabor wavelet is:

$$\Gamma(x,y) = e^{-\frac{5}{2} \left(\frac{x^2+y^2}{\alpha^2+\beta^2} \right)} \sin[-2\pi(U_\phi x + V_\phi y) - \psi] \quad (1)$$

Graphically these wavelets can be realized from the combination of a sine wave and a Gaussian envelope as shown in Figure 8. From observation of the spectral information in this figure it is

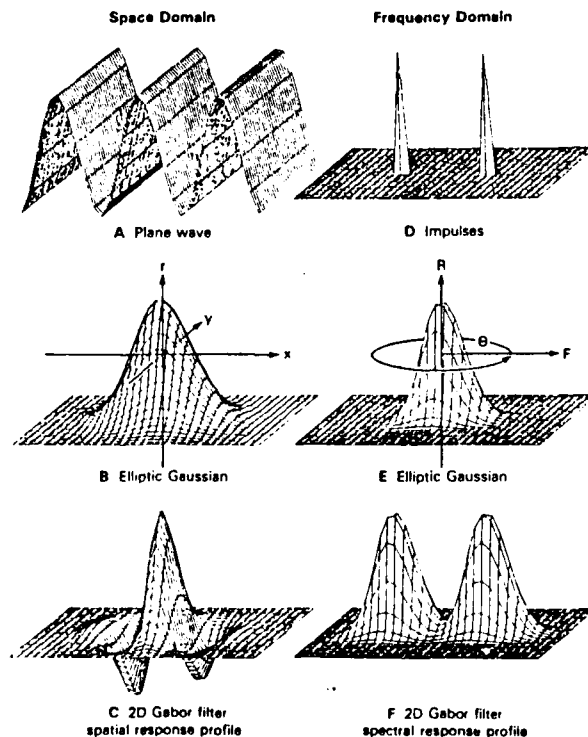


Figure 8: Construction of a Gabor Wavelet [Jones and Palmer, p. 1235]

intuitively obvious to the casual observer that in order to construct a complete set of these wavelets, rotations beyond 180 degrees need not be considered. The various mechanisms for altering the Gabor wavelet can be found in the variables of Equation 1.

The Gabor wavelet originated in a paper written by Dennis Gabor in 1946 [Gabor]. In this paper he discussed the need for a simultaneous time/frequency representation for signals. He then proved that this representation was limited in that ideal resolution could not be achieved simultaneously in both domains. As a result for any representation there would be a degree of uncertainty. To minimize this uncertainty he proposed a set of elementary filters composed of the product of a sinusoid and a Gaussian envelope. Gabor's emphasis was in the areas of communications and speech recognition; as a result the signals he proposed were single-dimensional [Gabor; Mueller et al.].

Not much was done with the Gabor wavelets in the areas for which they had been proposed. They were instead expanded into two-dimensional wavelets and applied to the analysis of the visual cortex [Daugman 80; Jones]. Positive results in this area and a simultaneous effort in expanding the mathematical basis for wavelets in general, have prompted a resurgence in interest in Gabor filters for communications and speech research.

Daugman developed a set of two-dimensional filters by expanding Gabor's elementary signals to a two-dimensional sinusoid which was then multiplied with a two-dimensional Gaussian envelope. Daugman was able to prove that the resulting two-dimensional wavelets, like Gabor's one-dimensional wavelets, represented an optimal space/frequency representation in that they minimized the uncertainty relationship [Daugman 85].

3.1.2. Gabor Wavelets and the Visual Cortex

Early explorations into the processes performed in the visual cortex gave evidence of a widely diverging set of possibilities. Work by a number of researchers developed a model in

which the most extreme portion of the posterior cortex provides a position-dependent representation of the image striking the retinal surface. This is Broadman's area 17 in the occipital cortex. Within this area are an enormous number of columns, each of which has the receptive fields of its cells mapped to a particular area of the retinal surface. These receptive fields are normally somewhat elongated and tend to increase in size as they radiate outward from the central foveal area, reflecting the reduced spatial resolution of peripheral vision (Figure 9).

Inside each of the columns are a number of pairs of simple cells which respond to events of different orientations. In the mammalian visual system these pairs are phase-related with a difference of close to 90° [Pollen]. The receptive fields of these pairs "must be conjugate pairs-that is one field with even symmetry and one field with odd symmetry around the same axis [Pollen, p. 1411]." Marcelja recognized that these could be modeled as Gabor filters with either sine or cosine under the envelope [Pollen; Marcelja].

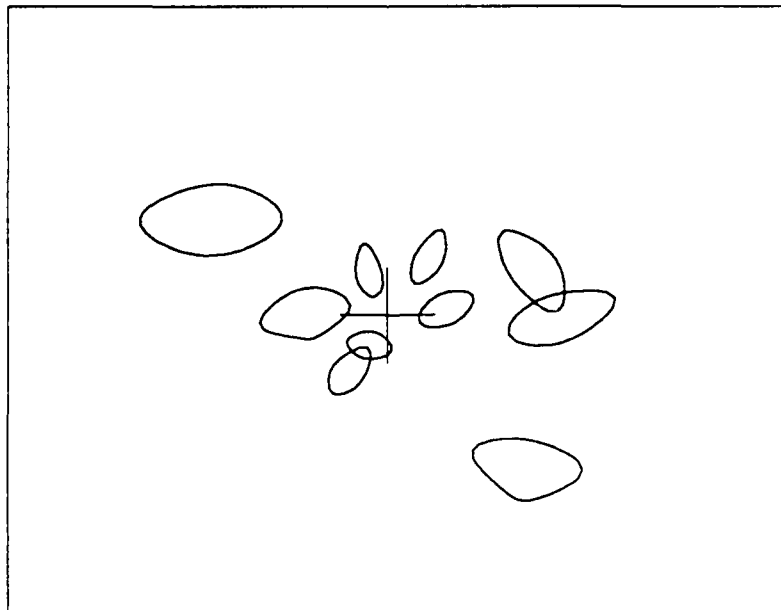


Figure 9: Typical Receptive Fields of Cortical Cells

Most early measurements of the properties used either solid objects or sine wave gratings. As a result of testing with bars and squares, Hubel and Wiesel felt that the cells of the cortex acted as edge and bar detectors [Hubel and Wiesel]. Others, testing with sine wave gratings, are just as certain that the cells were responsive to frequency events. These conflicts resulted from the propensity of the cells to provide much more information than researchers had anticipated. As a result, the cortex would willingly oblige the researcher by providing him with a response to match his input. In fact some researchers have taken the extreme position that cells "respond to aspects that might be thought of as teleologically important; that is, to moving objects which look like flies, to moving shadows which look like approaching predators, etc. [Ervin, p. 35]."

One possible method of exploring a system of this complexity is to force the system to characterize itself. A method for doing this is to drive the system with a grid of randomly activated uniformly distributed impulse functions and to measure the response of the system. To do this required a more complex setup than was generally in use.

In the early 1960s Ervin used a computer to provide an impulse display input to, and to record the responses of, receptive fields of simple cells in the visual cortex of cats. The analysis of the data was done partially on the computer, but to also to a large extent by hand. Still the plots of the data by Ervin presented a clear picture of the simple-cell spatial-response profile, although no mathematical description was fitted to this response [Ervin].

In later experiments Jones and Palmer used an impulse field to record the spatial response of simple cells in the cat [Jones and Palmer, 87a]. They also used drifting sine wave gratings to determine the spectral (time base) response of these cells [Jones et al.]. After the responses were recorded, a Gabor model was fitted to both the spatial and spectral data using the simplex algorithm. The fit of the model was measured by calculating the least-squared-error. The result was that no statistically significant error was found in 33 of 36 spatial responses and 34 of 36 spectral responses (Figure 10). Even in those cases with a statistically significant error, the level of error

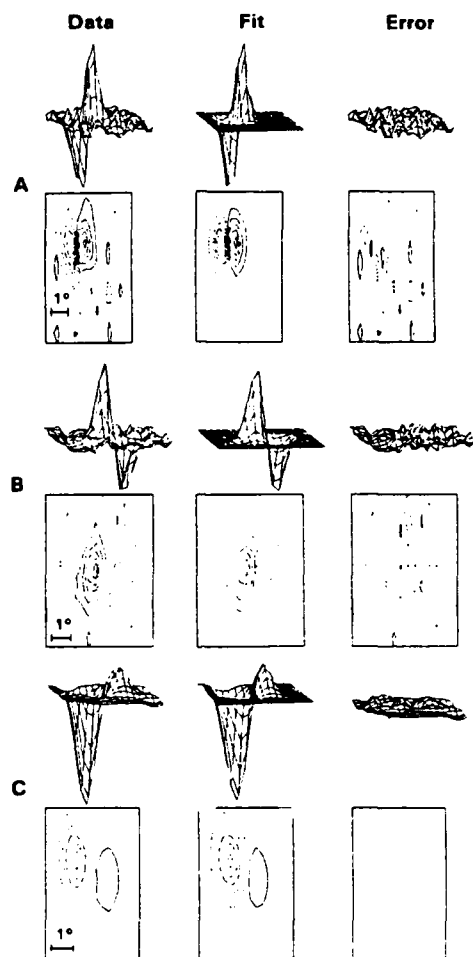


Figure 10: Comparison of Cat Cortex Simple Cells to Gabor Filters [Jones and Palmer, p. 1238] was just into the range of significance and observationally it appeared in a form similar to the filters used (Figure 11). This appears to confirm Marcelja's use of the Gabor filter model for simple cells [Jones and Palmer, 87b].

In addition to the simple cells, there exists a class of neurons in the visual cortex known as complex cells. These cells do not have the same response profiles as simple cells, and indeed tend to be quite non-linear in their response. They also tend to be directionally sensitive. Recent

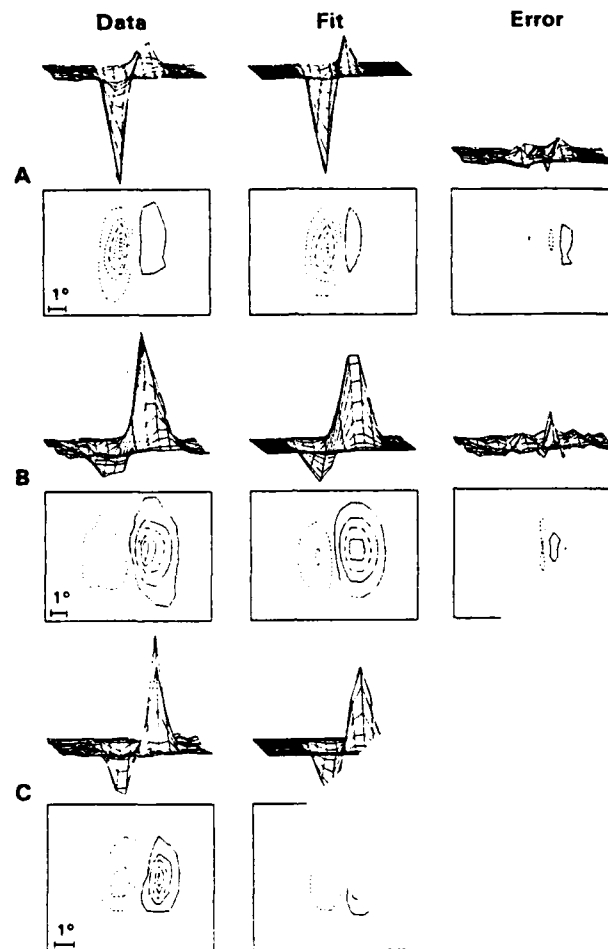


Figure 11: Comparison of Gabor Filters and Cat Cortex Simple Cells With Error
[Jones and Palmer, p. 1242]

investigations into these cells by Emerson et al. have resulted in a motion-energy model as a plausible description of the function of these cells. The form of this model is a two-dimensional Gabor filter, with one spatial axis and one time axis. Measurements on cats (Figure 12) have provided data to confirm this model, and a biologically plausible method for implementing the model with simple neuronal units has been demonstrated [Emerson et al.]. Although this model was described only in two dimensions Emerson has said that he expects that the actual model is

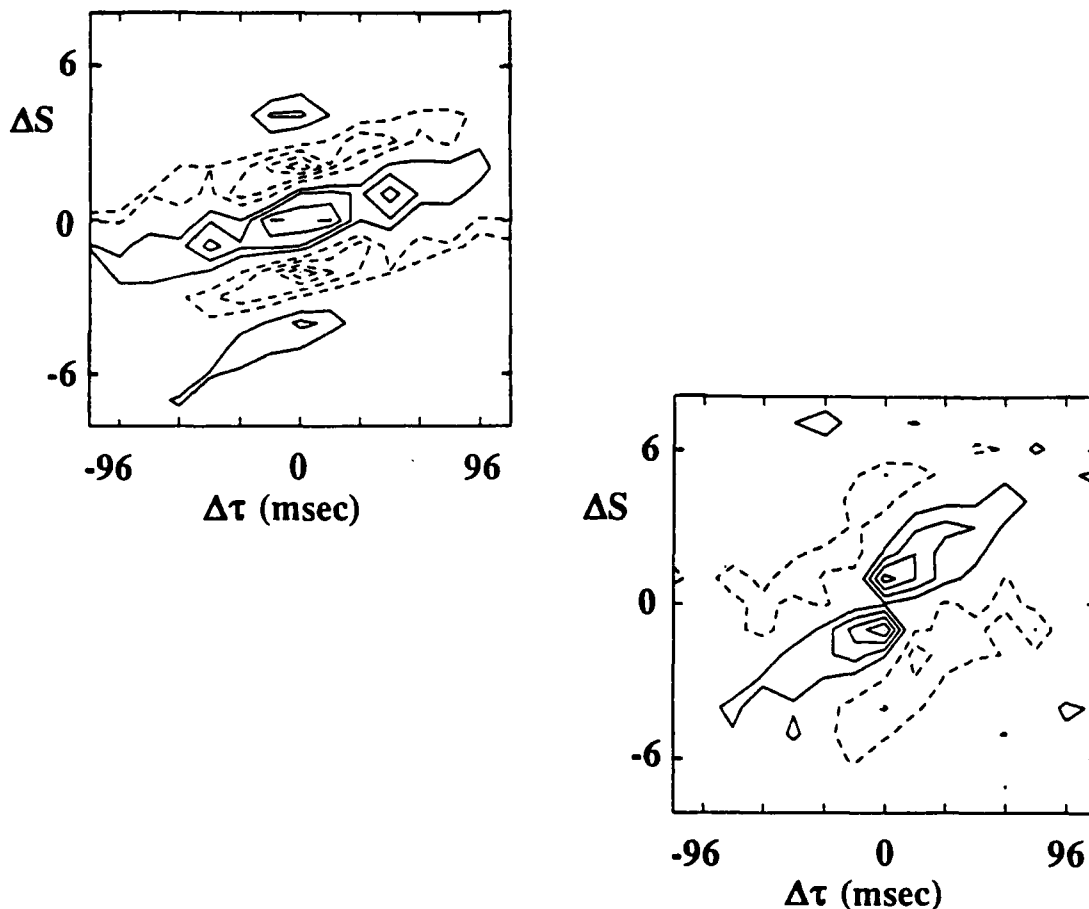


Figure 12: Complex Cells - 2-bar Interactions [Emerson et al, 90, p. 9]

"little elliptical cigar-shaped filters floating around in space [Emerson, 90]." Emerson's space has two axes of space and a time axis. This maps well to the two-spatial dimension surface of the brain. Figure 13 provides a graphical illustration of the boundaries of these filters. There is also evidence to support this type of a characterization in simple cells. Emerson and Citron [89] found support for this model in a two-dimensional space/time plot for a simple cell. Jones and Palmer's [86b] evaluation was in two dimensions of space; however, they do relate a certain time dependency to their data which they never analyzed in these terms. Taken together these would suggest that the same model may be valid for the simple cells and the complex cells; however for simple cells the central axis, or wave front of the sinusoid, of the three-dimensional Gabor filter

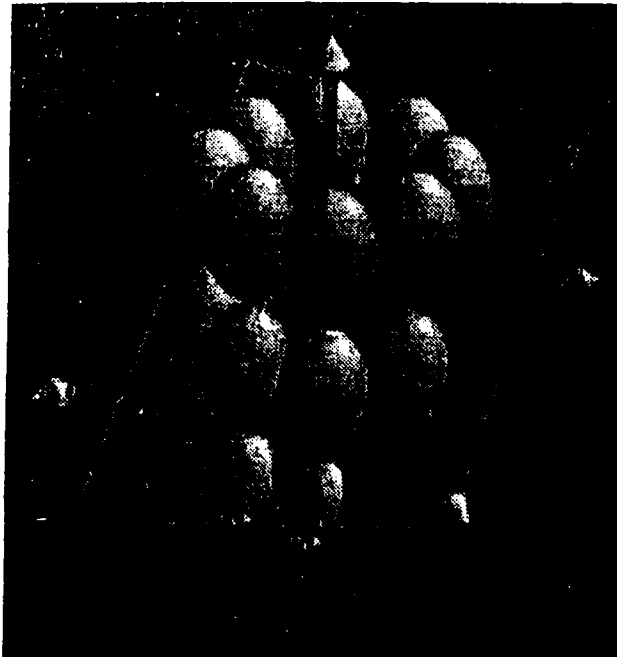


Figure 13: 3-d Filters [Citron et al., p. 184]

would lie more nearly parallel to the spatial plane. This is consistent with nature's standard practice of preferring to use simple, nearly identical units in a variety of ways to perform multiple functions.

3.2. Psychological Implications of Gabor Wavelets

Modeling the cells of the visual cortex has some implications about operations which could be performed using those models. It is reasonable to expect that the many of the same psychological phenomena observed in human perceptual responses should be able to be duplicated through use of the model. Among the most interesting of these phenomena are those images which give rise to illusions and other unexpected effects, for these expose the raw edges of the underlying system and are more likely to be dependent on the actual construction of underlying mechanisms

than the undistorted and consistent view of common scenes. The limitations as to how well the model can be expected to duplicate these phenomena is in many ways a measure of the fit of the model. Performing tests of these kinds on a model also provides the tie between psychologically observed responses and the underlying mechanisms which give rise to these responses.

3.2.1. The Simultaneous Contrast Bar

An image which is commonly used to demonstrate Mach bands¹ can be altered slightly to provide two additional effects. When a band of constant brightness of an intensity midway between the upper and lower colors is added to the image, two anomalies appear (Figure 14). In the first, the ends of the band appear to have different intensities, with the portion of the band in the white area seemingly darker than the end in the black area. This effect is called simultaneous

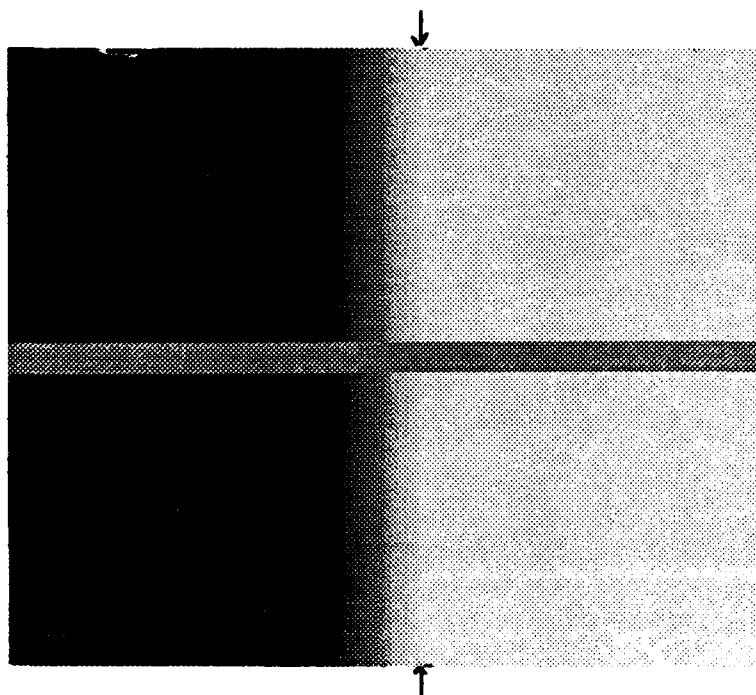


Figure 14: Simultaneous Contrast Bar

¹Mach bands are an illusion which occur along the borders of intensity slopes. At the very edges of the slopes, bands appear which are either lighter or darker than the adjoining constant-intensity surface. This effect can be observed in Figure 14. By placing a piece of paper across the image at the tips of the arrows, the bright band is made to disappear.

contrast. The other effect is located at the center of the band where it passes through the slope. There is a point in the ramp where the true intensity of the band is the exact true intensity of the band. What is of interest here is that the band and the ramp do not merge together at this point, but remain as separate and distinctive features. When casually observed the band in this area appears to be one separate and distinct feature. Yet if pressed, an observer, while tracing vertically down the line of constant intensity in the slope, finds it difficult to mark a distinct point where the intensities change from ramp to bar intensities. That is, he cannot identify with certainty the point where the pixels he is tracing become bar pixels rather than ramp pixels. There are seven distinctive regions in this scene - the upper and lower dark areas, the upper and lower slopes, the upper and lower light areas, and the band. The band is grouped into a single feature despite the appearance of distinctive colors on the ends because there is no point at which the colors can be separated. An informal survey I conducted has shown that some observers will group the scene into fewer regions; however, even these groupings will include the central bar as a single region. Examples of such groupings are: 1) Dark and light sides grouped together with a central region of changing intensity; the bar is not included as a region, but represents an edge. 2) The regions grouped as before with the bar forming a separate region. 3) The four corner areas as the only distinctive regions, etc.

The first process to be performed on this image is to filter it with sine Gabor wavelets. The magnitude of the resulting image is then filtered again. All wavelets have a horizontal orientation. The result is an image with edges along the whole length of the central bar. A line has been drawn along the detected edge (Figure 15). The top line plots the intensities of this line. Interestingly, where the bar crossed the slope in the original image, the edge is not strongly enhanced.

The next process to be performed is to filter the image with cosine Gabor transforms. The result is the image shown in Figure 16. The top line in this figure is a plot of the line drawn through the the center of what was the constant intensity bar. The plot clearly demonstrates that

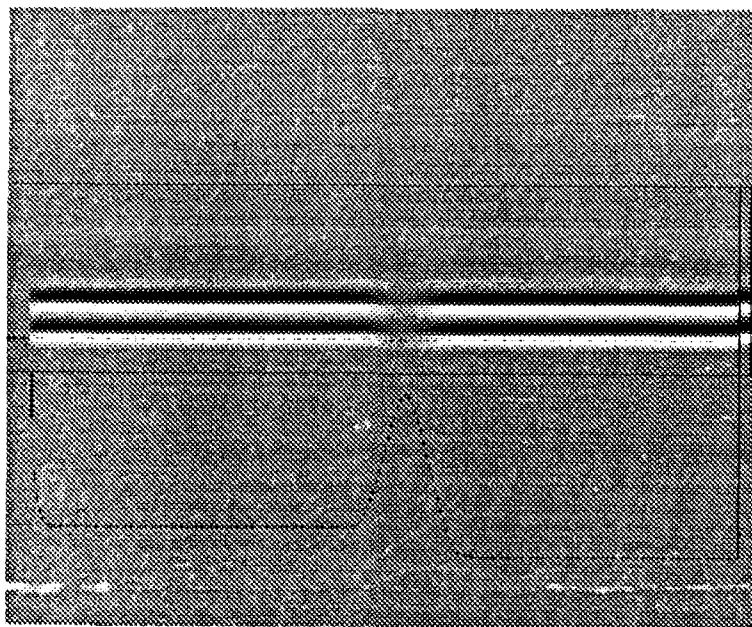


Figure 15: Constant Intensity Bar After Gabor Filtering to Enhance Edges

in the bar in this image does not have a constant intensity. This matches our perception of the bar in the original image.

A more interesting result can be obtained by using multiple Gabor wavelets. Figure 17 is composed of the maximum-intensity pixels from four Gabor filtered images. The filters used were: sine Gabor wavelets at 0 and 90 degrees, and cosine Gabor wavelets at 0 and 90 degrees. This figure reproduces all of the effects perceived in the original. Not only is the intensity of the portion of the formerly constant-intensity bar on the darker half of the scene brighter than the portion on the high intensity side, but the scene is also divided into seven regions with the upper and lower portions of the scene grouped by common intensities. Finally, though the bar is mapped as a single region, the edge between it and the sloped segments of the scene is somewhat indeterminate and it could be segmented into two regions if the pixel intensities were closely examined. There is a difference between the pixel intensities of the ends of the bar in this figure, although the difference is not so large as to make it readily visible. These results suggest possibilities for

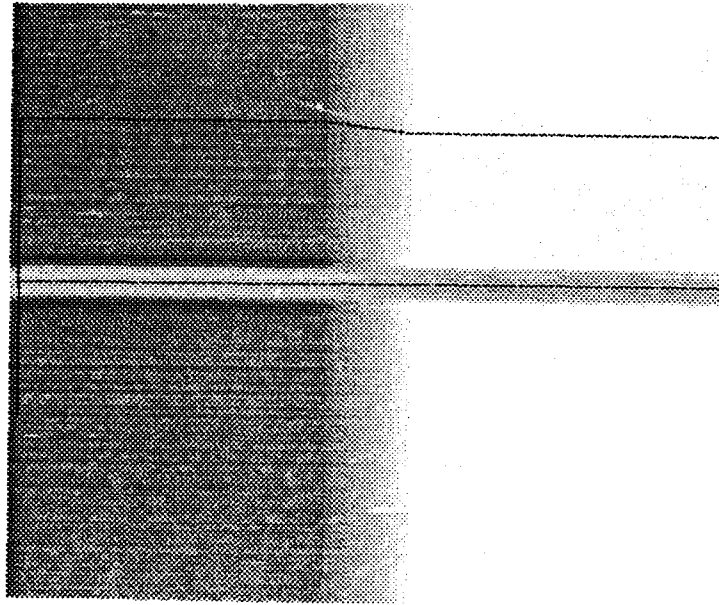


Figure 16: Simultaneous Contrast Bar After Filtering to Change the Bar Intensities

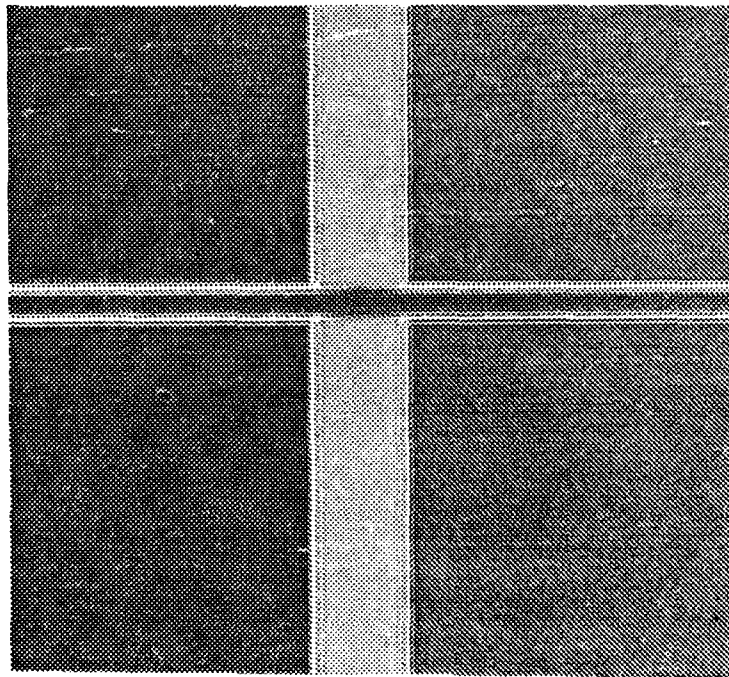


Figure 17: Combined Wavelet Filtering of the Constant Intensity Bar Image

important edge, region, and constancy effects being located in the visual cortex where Gabor wavelet-like cellular responses can be found.

3.2.2. The Spreading Effect or Assimilation

A common tool used by artists, newspaper illustrators, and cartoonists is the spreading effect, or assimilation. This is the illusion that a group of lines placed close to one another will appear as a single colored region. This illusion is demonstrated in Figure 18a. In this illustration the groups of lines appear as single regions. Correlating this image with a sine Gabor filter (Fig-

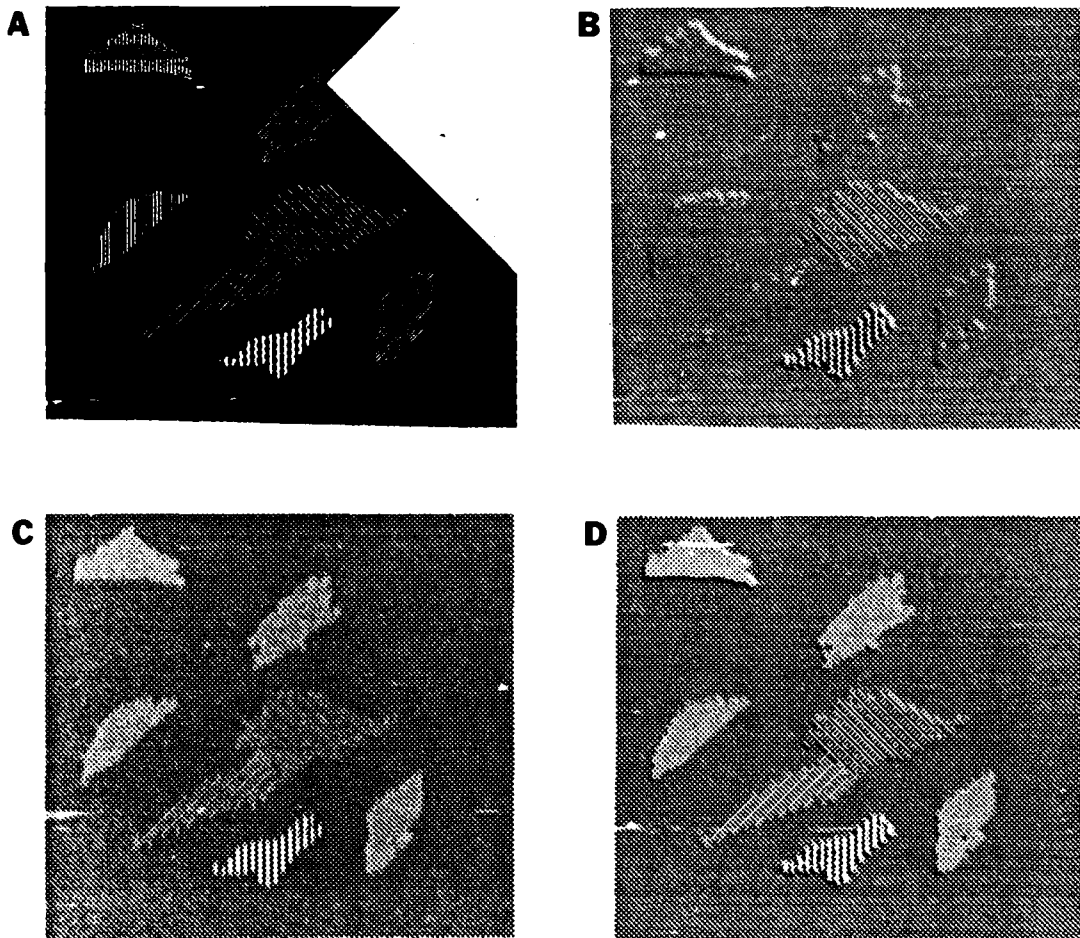


Figure 18: Assimilation of Closely Spaced Lines:

ure 18b) segments the image into areas defined by the extent of the line regions. Where the separation between lines is too large, or where there is strong texture in the image, the segmentation is not complete. Using a somewhat larger Gabor filter would improve the segmentation in these cases. This is the equivalent of holding the cartoon, or illustration, back a bit farther from one's face. The results of using a cosine Gabor function (Figure 18c) are equally impressive. In this case the regions outlined by the sine Gabor filters are given uniformly high values, which are dependent on the lines' width and spacing. Combining these results (Figure 18d) gives an image with the closely spaced lines assimilated and seemingly projecting outward from the image.

3.2.3. The Contrast Sensitivity Function

The contrast sensitivity function demonstrates the quality with which Gabor functions can be used to model portions of the human visual system. It is known that people are more sensitive to some frequencies than to others. People are increasingly sensitive to higher and higher frequencies, until the frequencies reach a maximum at about 3 cycles per degree of the visual field [Goldstein, p. 163]. After this point, their sensitivity rapidly declines. This can be tested with a sine-wave grating where the frequency of the sine-wave increases from left to right and the intensity increases exponentially from top to bottom (Figure 19). A trace of a typical human sensitivity curve is shown in Figure 20. The area under the curve is where most people are able to detect the grating. The sensitivity curve is somewhat orientation-sensitive; however there is no special selectivity for sine-waves at any particular orientation, such as horizontal and vertical sine-waves.

Figure 21 shows the result of correlating the sinewave grating of Figure 19 with a vertically oriented Gabor filter. The filter has the highest response to sinewaves with a period of about 24 pixels. The resulting image closely approximates the human visual response even to the limitations imposed by the rendering of the grating in an 8 bit grey scale. This is the scale of the

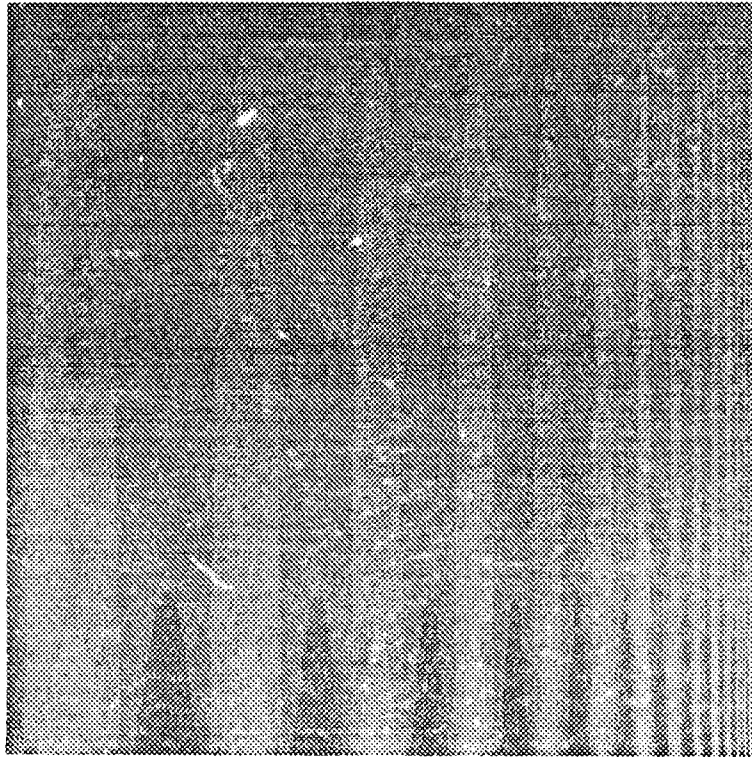


Figure 19: Logarithmic Sine-Wave Grating

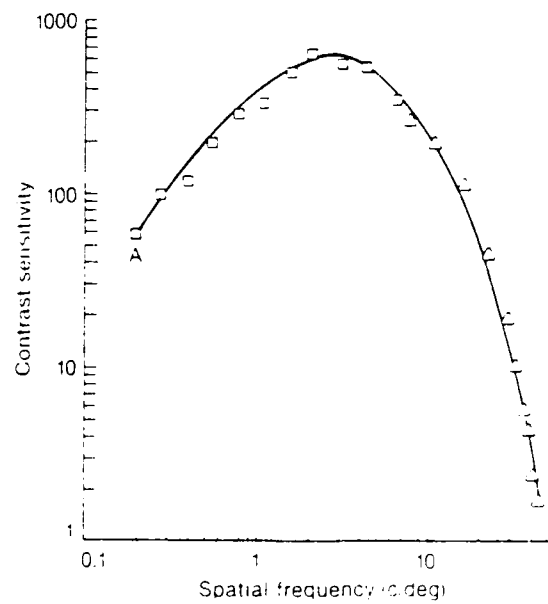


Figure 20: Human Sensitivity Curve to Sine-Wave Grating [Goldstein, p. 163]

figures shown in this text. At this scale, the grating, when closely observed, is still visible even at the upper portion of the image. This is because an 8 bit scale does not provide a wide enough range of intensity to escape detection of intensity edges. The Gabor filter also perceptibly detected the presence of the grating at the upper portion of the scale. If the image is thresholded to remove the effects of minor variations, a very distinctive curve, closely approximating the sensitivity curve of the human visual system, is observed.

3.2.4. The Muller-Lyer, Ponzo and Other Illusions

An interesting pair of illusions are the Muller-Lyer and Ponzo illusions (Figure 22). These illusions produce similar but opposite effects. In both illusions the horizontal bars are of identical

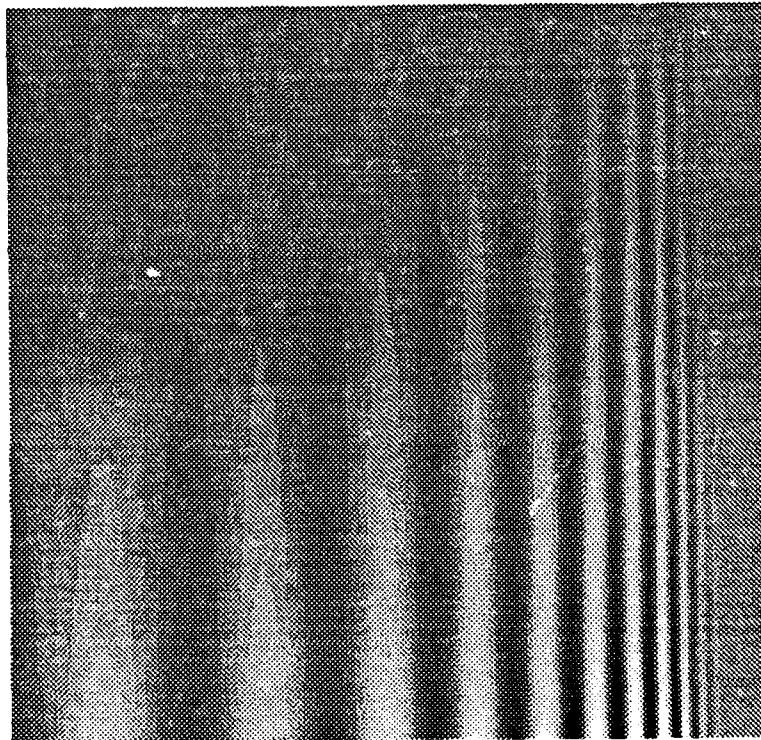


Figure 21: Gabor Filtered Sine-wave Grating

length. In the Muller-Lyer illusion the bar with the inward pointing arrows appears to be longer than the other bar. In the Ponzo illusion, the bar with ends closer to the outer lines appears to be longer. Interestingly, if the arrows are separated from the bars in the Muller-Lyer illusion (Figure 23), the relative lengths of the bars change and the illusion now approximates that of the Ponzo illusion. The observer now perceives that the bar with the outward pointing arrows is larger. This suggests that there are at least two different but opposing causes for the illusions. One illusion which extends lengths of bars when arrows are close the ends, and one illusion which extends the bar filling the largest amount of space between two markers.

Attempts to explain these illusions have generally been linked to the higher reasoning processes and overcompensation in attempts to maintain constancy in the environment. For example, objects should always maintain a constancy in their size - a dump truck is always larger than a pickup truck. Gregory proposed a theory that "size constancy normally helps us maintain a stable perception of objects by taking distance into account [Goldstein, p. 259]." In illusions the mechanism is misapplied, and clues surrounding similar-sized objects affect the way they are perceived, causing one to appear larger than the other. This appears to be reasonable in the case of

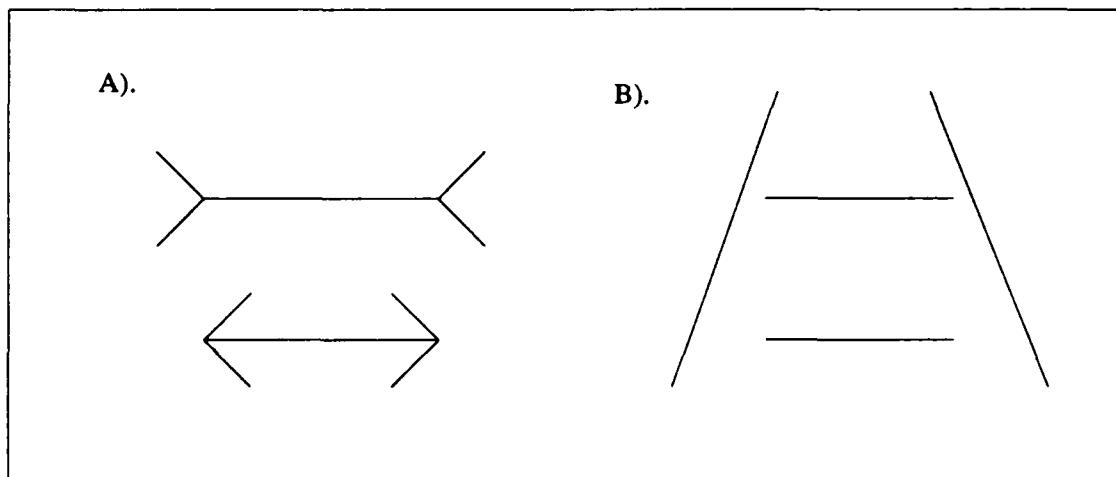


Figure 22: The Muller-Lyer (a) and Ponzo (b) Illusions

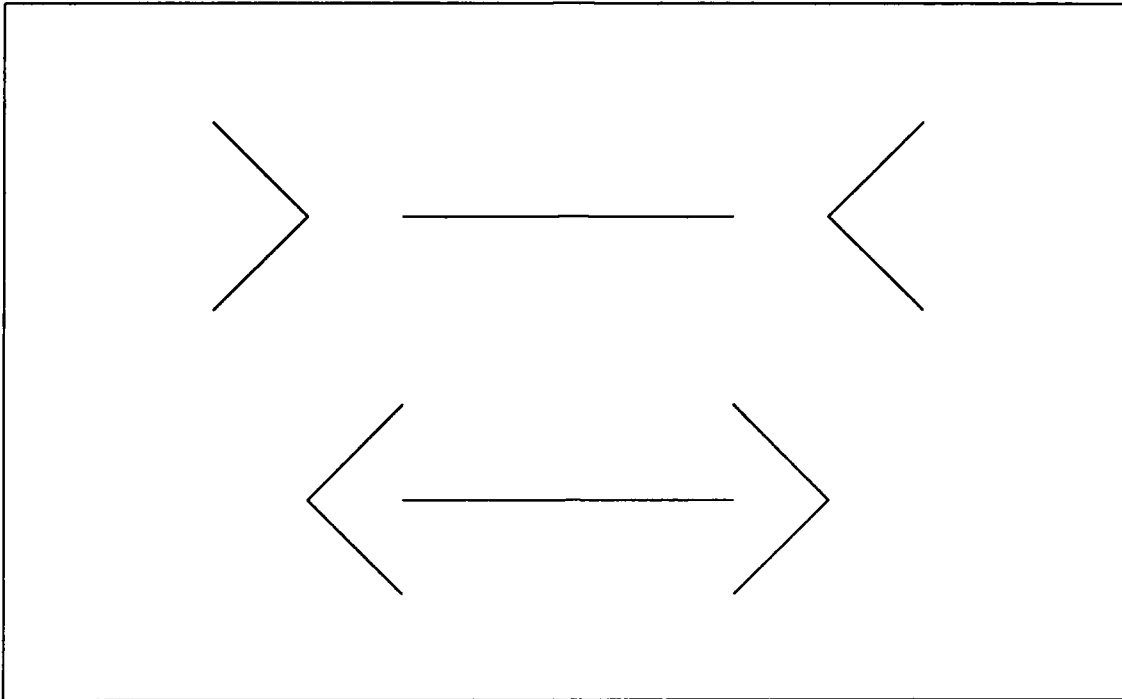


Figure 23: Modified Muller-Lyer Image with Illusion Reversed

the Ponzo illusion, but attempts to apply this theory to the Muller-Lyer illusion end in arguments that the lines represent the internal and external edges of cubes. This seems rather strained and fails to explain the reversal of the illusion when the arrowheads are detached. Another theory is that the illusions are the result of attempts at creating a three dimensional representation from the information available. However, versions of these illusions exist in which there are no ambiguous dimensional clues [Rock]. This implies that the illusions cannot be the result of attempts to add dimensional information which is not present in the image. Other explanations include: contour displacements, contrast and assimilation effects, and incorrect comparisons [Rock]. These are all stated to be the result of higher-level processing.

In approaching illusions from the lower-level processing side, one finds other attempts at explanations. One of these is the theory that illusions are the result of eye movements; however, the illusions are still present even if presented in too short a time for the eyes to move [Rock].

Further, evidence exists to show that the illusions are due to effects at levels of the vision system beyond the immediate vicinity of the eyes. "Most of the illusions can be achieved by fusing half images presented to the two eyes [Rock, Pp. 33-47]." Thus any process which is used in attempts to explain these illusions must occur after the inputs from the eyes have been fused together. Ginsburg theorizes that the illusions are the result of Fourier-domain filtering in the brain. However, the images he presents as evidence have entirely too many gratuitous effects to provide an adequate solution, though he is able to duplicate many illusions [Ginsburg]. One problem is establishing the biological ties for this solution. Ginsburg also overextends himself in claiming an explanation of the Ponzo illusion. Here his explanation is dependent on the upper bar's being so close to the outer lines that there is an interaction between them. Therefore, Fourier-domain processing does not appear to provide an adequate explanation for the source of illusions. Others have explained illusions as Laplacian filtering on the "Primal Sketch" [Shapely et al.]. This of course requires locating the "Primal Sketch" and a mechanism for producing the filtering.

A plausible biologically based explanation for some illusions can be found with Gabor filters. These filters model processes occurring in the striate cortex. Because of this they can be used to examine even those illusions which can be created through the fusion of two partial images. Gabor filters also provide the frequency limitations sought by Ginsburg, and do not rely on high-level processes. If the Muller-Lyer illusion is processed by convolving it with a horizontally-oriented sine Gabor filter, the result is a physical reproduction of the effects perceived in the illusion (Figure 24). The bright area of the filtered illusion is larger for the inward pointing arrows (top of image). This is, in fact, not an illusion after filtering. Thresholding the image shows that the length of the line with a brightness of greater than 200 (on a scale of 0 to 256) is 148 pixels. The length of the lower line is 136 pixels, an 8% difference in length. Nor are there clouds of extraneous effects. Gabor filters can also be used to extend the length of the upper bar in the Ponzo illusion (Figure 25a), as long as the upper bar is within a distance from the outer

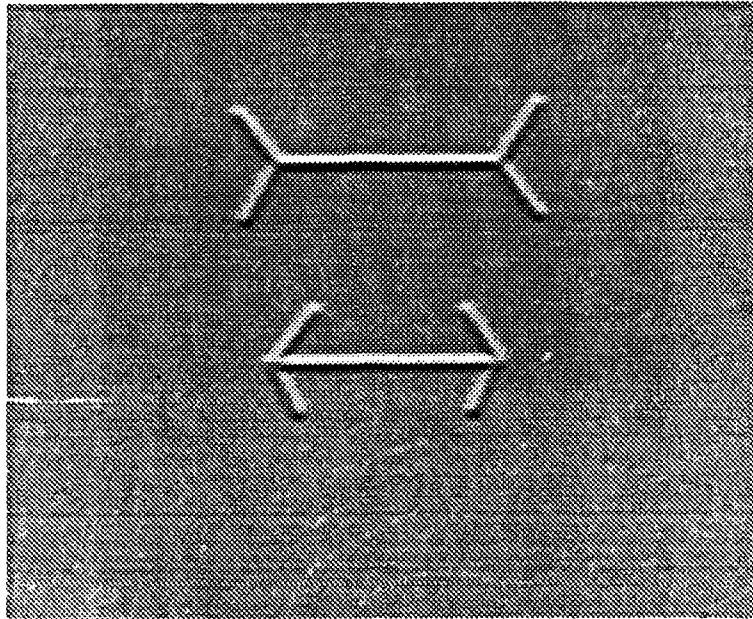


Figure 24: Gabor Filtered Muller-Lyer Illusion

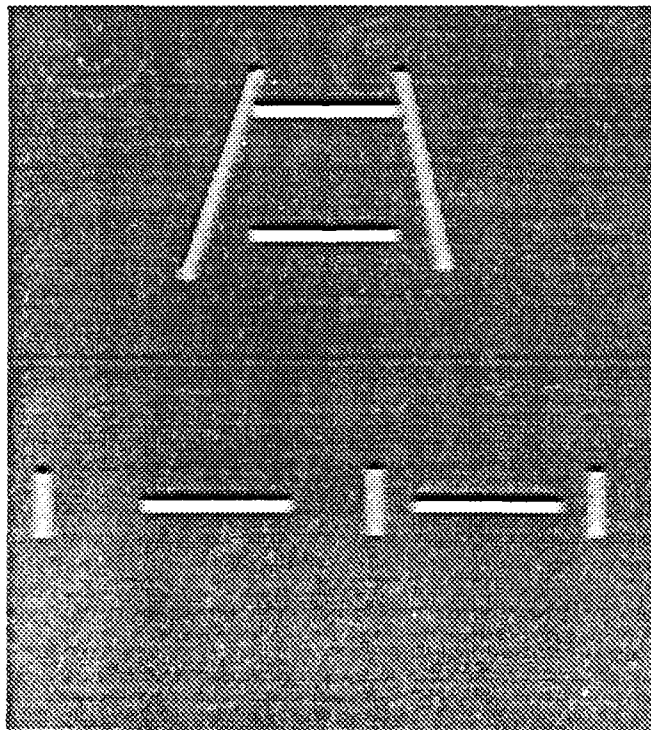


Figure 25: Two filtered Ponzo Illusions

lines equal to, or less than, the size, or dilation, of the Gabor filter. In the bottom example of the illusion (Figure 25b), there is no way to duplicate the illusion via Gabor filters. This suggests that while Gabor filtering may provide a partial explanation, there is in fact some other, potentially higher level effect occurring.

Another illusion which can be duplicated using Gabor filtering is the Poggendorff Illusion (Figure 26a). The illusion is that the diagonal lines would not meet if extended, even though in reality they would. When viewed closely, the vertices of the filtered image reveal that the lines do indeed diverge, with the slopes of the lines changing as they approach the vertical lines (Figure 26b). The illusion, and its Gabor-created duplication, persists for the obtuse angles alone (Figure 26c), but not for the acute angles (Figure 26d).

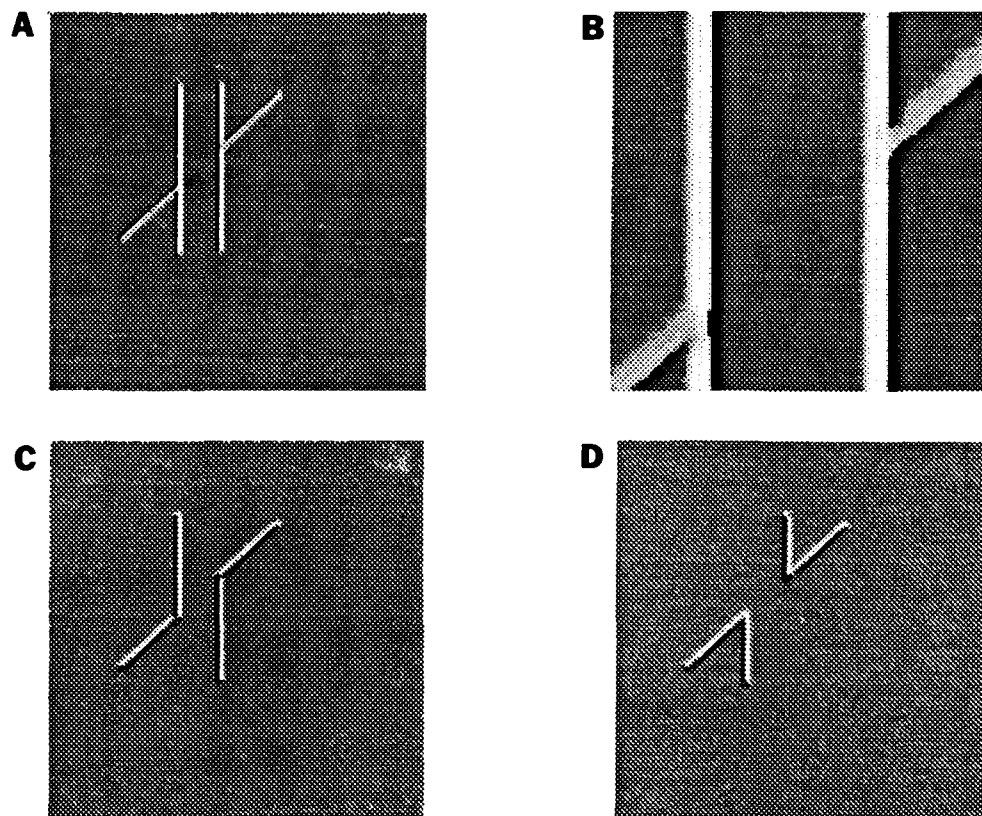


Figure 26: Aspects of the Poggendorff Illusion

The Zöllner illusion is created when parallel lines are covered in cross hatches. These cross hatches are tilted at opposing angles on alternate lines. The result is a perception that the lines are no longer parallel. When the Zöllner illusion (Figure 27a) is convolved with Gabor filters, the line segments between the cross hatches shift their orientation toward the normal of the cross hatch. The result is that the segments within the lines are in fact no longer parallel (Figure 27b). Measuring the ends of the lines in the Gabor filtered image shows that in the overall perspective the lines remain parallel. The overall illusion seems to be dependent on both localized changes such as those introduced by the Gabor filtering, and on a higher level, more global mechanism. This is very similar to the effect induced by Laplacian filtering on the Münsterburg, or cafe-wall illusion [Shapley et al.]. In this illusion parallel mortar lines on a wall with a checkerboard pattern appear to converge and diverge.

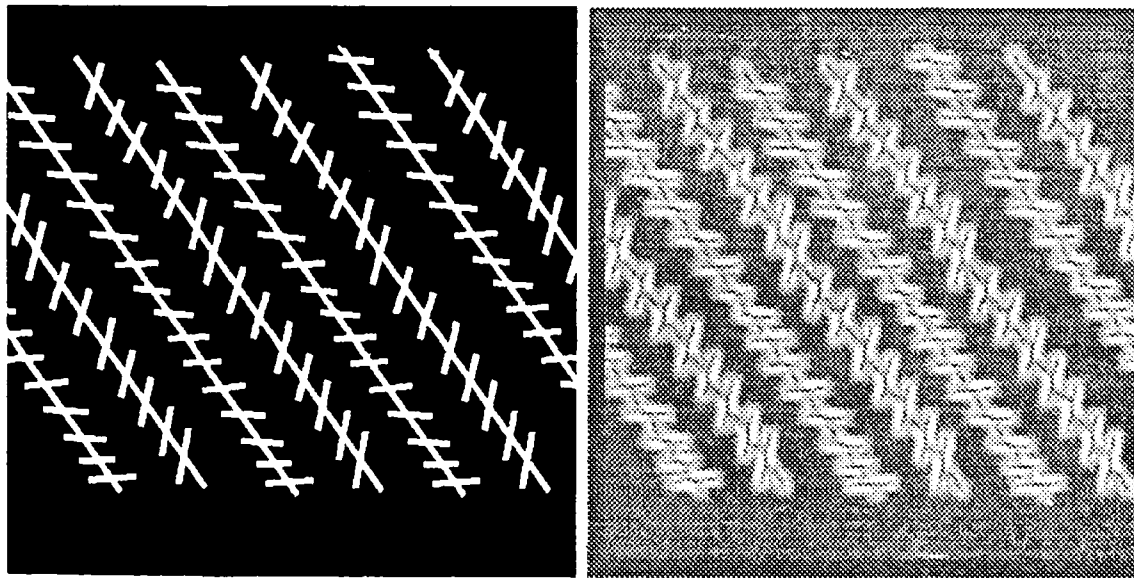


Figure 27: The Zollner Illusion

The Wundt illusion (Figure 28a) gives the impression that the edges of a square superimposed over several concentric circles tend to bend inward toward the center of the circles. When this image is convolved with a horizontally-oriented sine Gabor filter, the result is a stair-step effect along the upper and lower edges of the square (Figure 28b). The overall line remains straight with no sag in its center, but the segments of the line where it crosses the circles bend to form the steps. This same effect can be seen in the vertical edges of the square if the proper filter is used. This stairstep effect is much like that found in the Zöllner and Münsterburg illusions. The global properties of the lines in all three of these illusions are not altered, but the local properties are effected by Gabor filtering in a manner which gives the perception of change. We expect stairs to lead up or down and we also expect stepped lines to change their level. The fact that many illusions cannot be globally duplicated, but are amplified by the filtering and by the local changes filtering produces, again suggests that there may not be any one cause for the

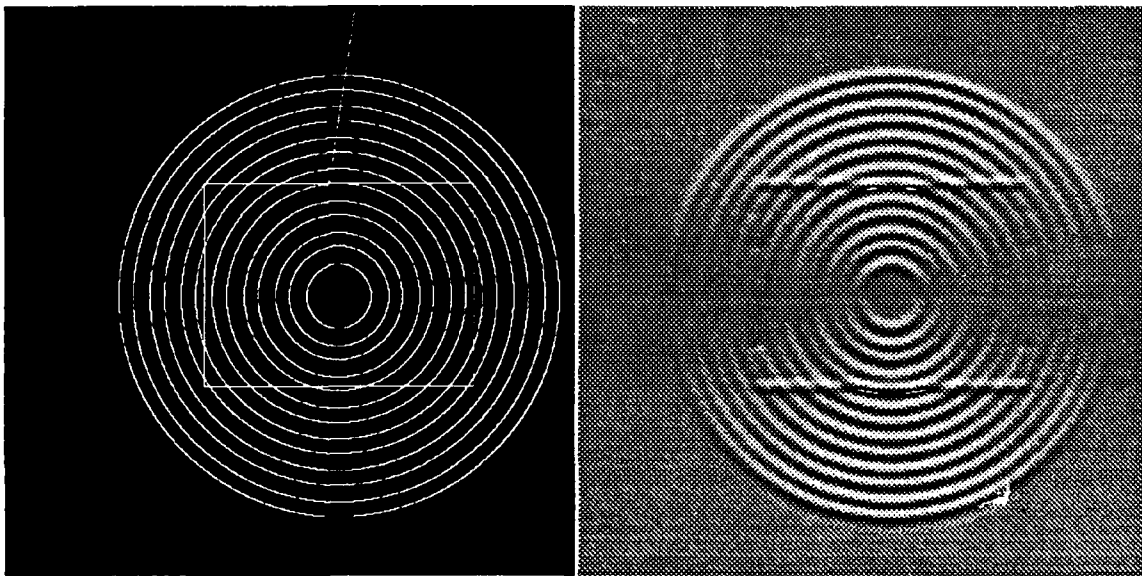


Figure 28: The Wundt Illusion

occurrence of optical illusions, but rather that low-level and high-level mechanisms combine to produce the effects.

Undoubtedly, there are numerous other illusions which can be duplicated or enhanced through Gabor filtering. Gabor wavelets have also been used to explain such visual events as Mach bands [Fiorentini et al.], texture detection [Turner; Daugman 88; Bovik et al.] and motion [Heeger; Adelson and Bergen]. The important concept here is not so much whether Gabor filters are in fact the actual structure used in the brain, but rather that so much of the activity and responses of the visual system can be accurately modeled using these tools.

3.3. Applying Gabor Wavelets to the Vision System Model

The preceding sections have shown how Gabor wavelets can be used to model portions of the human and mammalian vision systems. This modeling has been done by others who have fit Gabor filters to measurements of cellular responses, and by our modeling of optical illusions. The effectiveness of this modeling is seen in the closeness with which processing optical illusions with Gabor wavelets can provide an approximation to the way we ourselves perceive these illusions. From these data, we can deduce that Gabor wavelets can provide a useful tool within our vision system model. In the next sections we will describe the use of Gabor wavelets for input transforms - for finding edges, and as a preprocessor for feature vectors for a backpropagation decision network - and for attention mechanisms. Because of the number of uses which can be obtained through the use of Gabor transformed images, it can be useful to view them in terms of a sensor transform, which is then used as input to a variety of input transforms and to the attention mechanisms. One interesting application of Gabor wavelets is as basis functions in a recognition/reconstruction network. This network, the pseudo-neocognitron, could serve either as an input transform, or at a higher level as a model for the super-conscious of a vision system.

3.4. Edge Location

One of the unique aspects of the use of the Gabor transform on images is its inherent directionality. Gabor filters ring only on features which are aligned with, or close to, the axis of origin of the sinewave of the filter. This feature, which has been exploited to some extent in finding objects, can also be used to provide information about the locations and directions of edges.

The most direct way in which to use Gabor filters to find edges is to simply correlate the image with Gabor filters at significant orientations. Edges which are in line with the axes of origin of the sinewave components of the Gabor filters will show a significant response. Edges with other orientations and areas without edges will show less response. The correlation planes can be thresholded to provide the edge information. If the locations of all edges need to be provided at once, the correlation planes can be combined. One method for combining the planes is to do so by picking the most significant values for each pixel and placing them into a new image. Other methods, such as logical operations on the correlation values or morphological operations, may also be effective.

Using Gabor correlations differs from most conventional techniques in that it does not only consider a single pixel and a few of its nearest neighbors, but also includes the entire context in which a pixel is located. This is important because it takes into account the fact that edges are not entirely local events. Another advantage to the use of Gabor correlations is that it is sensitive not only to step edges, but also to sloping edges.

The direct use of simple methods for obtaining edges from Gabor correlations can be effective for many uses, but it has some limitations. In images with a large amount of noise or many local texture edges, the true edges can be lost in the process noise. These methods also don't always isolate a line to a single pixel width, and it is difficult to find a threshold which finds the edges and does not accept false alarms. Therefore more elaborate methods are needed.

Burns et al. proposed a scheme which used gradients to find straight lines. In this scheme the gradient orientations were calculated by using a number of small (2 X 2, 1 X 2, etc.) operators. The resulting vectors were then grouped into edge-support regions of common orientation. The edge-support regions could then be fitted with lines. If the regions were too small the support for a line was discounted. Likewise, edge support regions could be tested for the gradient steepness [Burns et al.]. The concept of introducing the use of gradients is an important one, as is the concept of looking at a larger scale to determine where the edges are. However, the techniques used in grouping the line-support regions and fitting edges to them appear to be somewhat more global and analytically involved than might be expected from a biologically oriented model.

Gradient orientations can be derived from correlations with Gabor filters almost as a by-product. This results from the orientation of the sinewave component of the filters. Further, the spatial extent of the Gabor filters allows them to consider not just a single point, but a local region in determining the image gradient for any given point. Gradient vectors can be established by using a set of Gabor filters with orientations which extend from 0 to 180 degrees. The gradient orientation is determined by selecting the highest-responding correlation at every point. The fineness of the orientation is determined by the size of the set. The sensitivity to local edges is selected by the size of the Gabor filter fields. Smaller fields are more sensitive to local edges. This type of a system is a biologically plausible model. Hubel and Weisel among others have located orientation-columns in the visual cortex [Hubel and Weisel; Goldstein], and Suter and Kabrisky, among others, have demonstrated the ability to construct a neural net which picks maximums [Suter and Kabrisky].

Although biologically plausible, constructing gradient orientations from a large set of Gabor filters is not computationally efficient. A far better approach is to use two orthogonally oriented Gabor filters. Responses to these will cover the space of possible orientations. The

specific orientation can be determined by calculating the arc-tangent of the responses to the orthogonal filters. The results combine into a new image called a gradient flow diagram (Figure 29). The gradient flow diagram shows the orientations of all slopes in the image. It includes gradient information for edges which are extremely weak, as well as for strong edges and even flat areas. A means is needed for determining which gradients in the flow diagram are significant. This can be done by combining information about the strength of the Gabor correlation along with the gradient direction.

A simple method to determine significant gradients is to use a modified flow diagram. This diagram is calculated by using the difference between the absolute values of the horizontal and vertical Gabor correlations rather than the arc-tangent (Figure 30). An edge is assumed to be significant if the values on either side of the edge differ greatly. Thus the most significant gradients are those between regions of light and dark areas. A positive value (light) in this diagram represents a horizontal edge, a negative (dark) represents a vertical edge. The areas with little indication of either a horizontal or vertical edge lie near 0 (grey). This modified flow diagram is useful in images which have primarily vertical and horizontal elements, or where these elements represent the features of interest.

Once the gradients have been identified, the edges and regions need to be extracted from the image. This can be done either by extracting the regions directly using split and merge algorithms [Querns] or by extracting edges and filling the regions between them [Fretheim]. Either method requires the establishment of criteria for region boundaries. Such criteria could be best fit of lines to a region [Burns et al.], changes in gradient direction, or the midline of a region of common gradients. Of these techniques, only the first guarantees that the lines will be straight. Changes in gradient direction are effective when the lines in a image are roofed lines (Figure 31a). The changes will occur along the top of the ridge formed by the line and at either side. The side lines can be filtered by requiring that the lines to also have a high Gabor correlation. On step

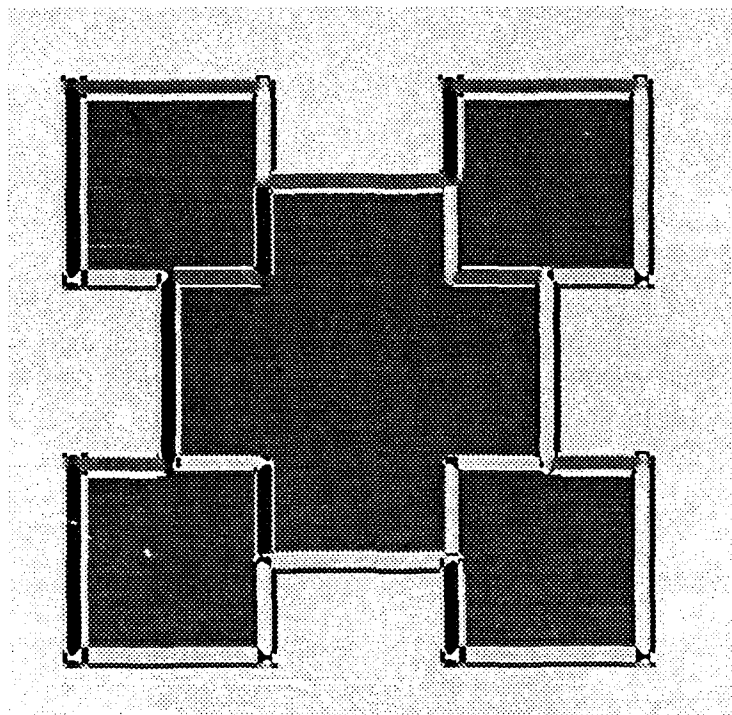


Figure 29: Gradient Flow Diagram

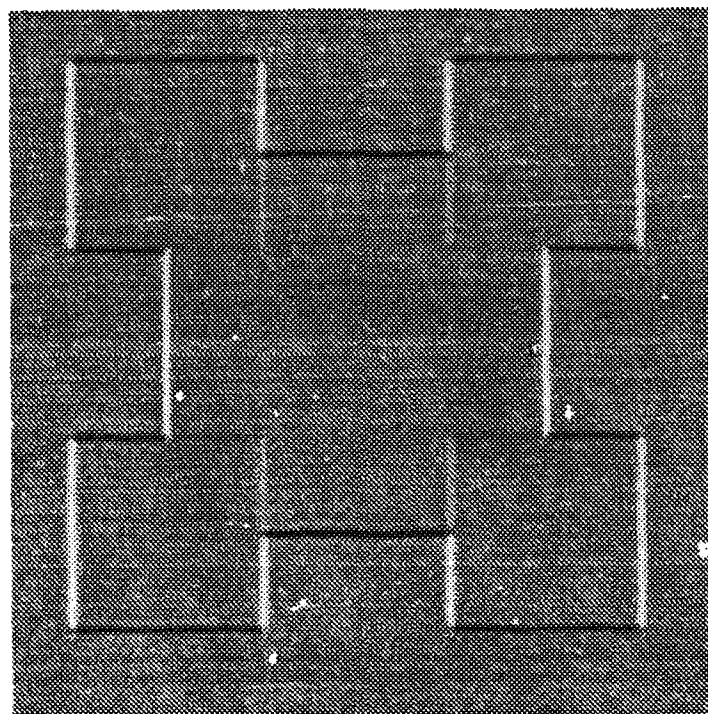


Figure 30: Image Gradients Coded as Difference of Horizontal and Vertical Gabor Correlations

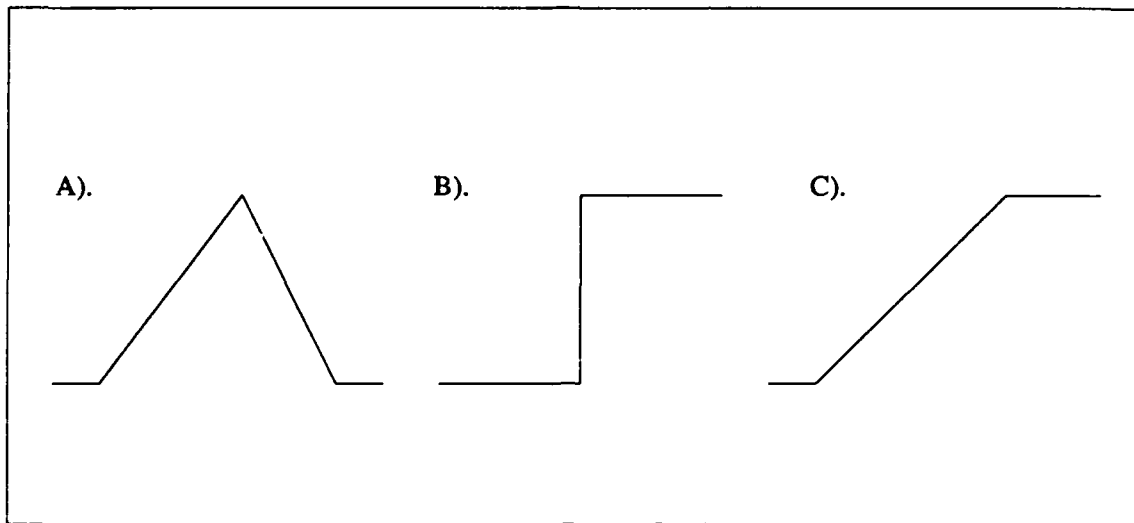


Figure 31: Edge Types: A) Roof Edges. B) Step Edges. C) Slope Edges.

edges (Figure 31b) this technique will create two lines, one along the top of the edge and one along the bottom. It may create even more, as step edges can create low-level artifacts in the Gabor correlation where a parallel Gabor filter can obtain a higher correlation coefficient than one oriented perpendicular to the edge. The result is extra responses oriented parallel to the real edges. Trying to filter these edges by using the degree of correlation can also result in the removal of the desired lines, as they are somewhat displaced from the actual edge locations. Sloped lines (Figure 31c) can also be susceptible to these problems, although they are not as likely to include extraneous artifacts. The displacement of edges is even more prevalent in the sloped lines.

Using the midlines of regions of common gradients places the extracted step edges in their proper locations. The method places edges through the middle of continuous gradient slopes. This provides an accurate estimate for sloped edges. Roof edges will be represented by two edges, one through either side of the edge. Problems encountered by this method include the extraction of lines in response to artifacts in the flow diagram, and in the centers of flat regions. These problems can be resolved through comparison of the extracted line with the correlation

coefficient. As the correctly extracted lines are in their proper places, they will not be summarily deleted; the artifactual lines will.

3.5. Object Identification

One of the interesting divisions of the functions of the human brain is the use of separate areas for direction of attention to, and for identification, of objects. Although damage to the attention related areas of the brain prevents a subject from locating and directing attention to objects, it does not prevent him from identifying objects. The recognition process proceeds on a parallel pathway [Goldstein].

One of the areas which has been identified as playing a major role in the identification process is the lower temporal lobe. In pathological experiments, if the temporal lobe has been damaged, the subject is able to locate objects, but is unable to identify them. In some cases, the subject is able to draw out the details of what he sees, but is unable to name the object [Treisman; Goldstein]. This is obviously not the only area in which identifications are made, as damage to other areas can cause failure to recognize objects as well. It is the difference in the extent of the inabilities which is interesting. With damage to the inferotemporal lobe, the loss appears to be a complete inability to compose an identifiable structure, but with damage in other areas the losses seem to be more specific - faces, color, motion, etc. [Luria; Treisman]. While the data are very sketchy and incomplete, the suggestion is that there is an area responsible for constructing the visual system's data into a unit for recognition.

The type and sources of data for identification may include many things; among them are motion, shape, size and color. These appear to be processed through separate paths, although they may all be used together for identification purposes. Some of the pathways appear rather clear. In the visual areas of the cortex there are indicators that particular sections - layer 4B, the blobs, and the inter-blob areas of area 17; and the thick-stripes, the thin-stripes, and the inter-stripes of area 18 - are responsible for the processing of different types of visual information

[Treisman et al.]. Yet all of this must be combined and fed to a recognizer. The problem is how to extract relevant features and encode them in a compatible format.

Extracting relevant features is not a trivial task. The selection of features can in fact be the key to the whole recognition task. Were it not it would be possible to simply feed the video output of an object directly into some type of recognizer and the identity would be immediately established. This does not happen, at least not with any non-trivial object. Thus many different features have been used, from simple size and intensity measurements to Zernike moments and other such complex choices. While each has a use, they do not generalize. What works in one problem does not work in another.

Biological vision systems do have a capacity to generalize. It may be the very limited capacity of the frog to detect several types of flies, including some he has never seen. Or generalization may take the form of the human ability to recognize a seemingly limitless number of objects. This generalization suggests that there is some feature set the performance of which is, if not universally perfect, at least adequate for most situations. Some biologically inspired candidates to serve this function, or to provide at least a portion of the set of features, are the Gabor-like functions found in the visual areas. These have been shown to be able to encode a number of different feature types, to include motion, color, orientation, texture, etc.

Another biologically inspired model is the back-propagation decision network. The back-propagation network, like all connectionist models, uses a large number of highly interconnected simple nodes to perform complex tasks and in this respect, at least, seems to imitate some aspects of neural connectivity. In the back-propagation network the nodes are arranged in layers. The nodes in each layer are densely connected to each of the nodes in the layer below and above it. The links between nodes are assigned weights. There are three types of possible layers: input layers, output layers and hidden layers. Nets can be constructed with any number of hidden layers.

Back-propagation networks function in an iterative manner. A feature vector is presented to the network at the input nodes. The values for the features are weighted and passed along each of the forward links. At each node, the weighted inputs are summed and some non-linear function is used to determine the output of the node. This output value is then weighted and passed along to each of the nodes in the next higher level. If the net is being trained, the values of the nodes at the output layer are compared to the values for a "correct" response. The error in the response is calculated, and the weights of the connections are updated. The error is then propagated back to the next lower level. Here, an error estimate is again calculated using the estimates from each of the connections. This error is used to update the weights of connections at the node and is then propagated back one layer further. After all of the weights have been updated, the cycle is repeated. The forward/back propagation cycle is repeated until the error in the outputs has converged to some acceptable level. This may take 100,000 or more iterations, depending on the problem to be modeled by the network. When the error has reached this acceptable level, the net is considered to be trained. The back-propagation path of the network can then be turned off, or left on. The advantage to turning off the learning portion of the cycle is that the network is then locked in and its learned responses will not drift even if the data are presented in large homogeneous blocks. The disadvantage to turning off the learning is that the network is not able to adapt to changes in the inputs, or to differences between training and test data.

The forward weights of the back-propagation network become models of the data the network is trying to classify. The output nodes provide a measure of the correlation between the inputs of the network and the model which the network has built internally for each output node. If the network has been properly trained there will be one, and only one, output node with a maximal response for each input. If the training data are clumped in groups it is possible that the network will not simultaneously build its model for all of the possible inputs, but rather for each in

turn. In this case, the network will not function properly. There are several other pitfalls which also need to be avoided. These are discussed fully in the literature and will not be covered here.

The prime advantage of the back-propagation network is that once the network has been trained, it is possible to get a solution for any input feature with only one pass through the network. This represents a significant speed-up over conventional techniques for networks directly implemented in silicon. As back-propagation networks are not now so constructed, and are likely never to be², they must rely on their other advantages. The robustness of the back-propagation network is important. The network degrades gracefully; the loss of each node only slightly degrades the overall performance of the network. Back-propagation networks are also great at interpolating data points which lie between those which they were trained for. These advantages, and the ability to train the network by feeding in the feature vectors without having to try to interpolate the key data points, make the back-propagation network a useful tool.

3.5.1. The Pseudo-neocognitron

The back-propagation network does have a serious limitation in that when it is set up to classify an object, it cannot provide a reconstruction of that object. There are other associative networks which can provide a reconstruction of an object from a partial object, or which can provide both a classification and a reconstruction. The problem with these is that the reconstruction they provide is that of the "pristine" object. That is, the object is not reconstructed as it actually appears with all of its deformations intact, but rather as the perfect object which the network has stored. The reconstruction of pristine objects is all right if all that is desired is a classification and a pretty presentation, but it is wholly inadequate to tackle the real-world problem of placing the object in context. Real-world objects are rarely pristine, and exactly where their constituent pieces and their deformities lie is important.

²One researcher has calculated that to provide a mildly interconnected network would require a surface area of 86 *meters*² [Bailey].

One network model has been proposed to address the problem of reconstructing objects in their true form. This network is the neocognitron [Fukushima]. The neocognitron uses a layered structure (Figure 32). Each layer is constructed from two types of forward-path cells and two types of back-path cells. The first type of forward-path cell, the U_i cell, is grouped in sets which perform identical calculations, spatially offset on the layer's input plane. Through a system of lateral inhibition, the output of the highest-responding U_i cell is passed to the U_c cell of the layer. As the activations progress upward through the network structure, each cell builds its recognitions from cells which cover larger and larger areas. As a result, the recognition is adjusted for features which are displaced at lower levels. The final recognition-layer is a set of cells which each respond to a particular input pattern.

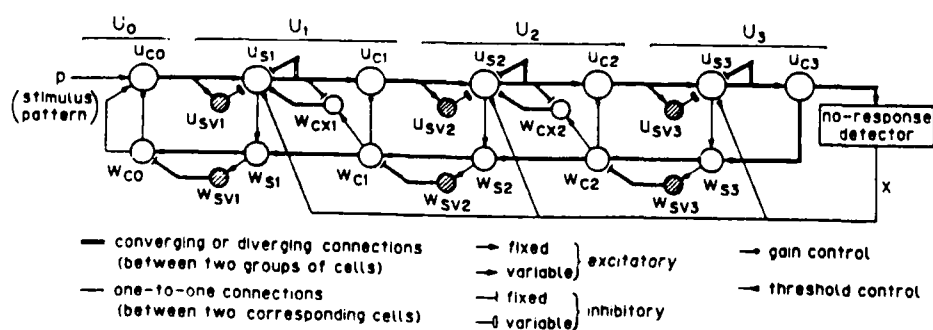


Figure 32: Structure of a Neocognitron [Fukushima, p. 4986]

After an object has been recognized, a backward path is activated. This backward activation spreads along the path of the highest-responding cells. This allows the input pattern to be reconstructed at the lowest level with its deformities intact. The backward activation of the neocognitron also serves to reinforce the forward activations. In doing so, the network strengthens its response to the recognized image and allows itself to detect the remaining features of the object, even if the features are only weakly present. This is done with the aid of the connections from the W_c cells to the U_c cells, and through the connection of the W_{cx} helper cells.

The other type of a cell included in the network is the U_{sv} helper cell. This cell is used to calculate the mean energy of the inputs to the U_s cell:

$$U'_{sv}(n) = \sqrt{\sum_{k=1}^{K_s-1} \sum_{v \in A_s} c_l(v) [u'_{cl-1}(n+v, k)]^2} \quad (2)$$

[Fukushima]

where $c_l(v)$ is a monotonically decreasing function of $|v|$. This energy is then used to normalize the value of a correlation function within the calculation of U_s :

$$\tilde{U}'_{st}(n, k) = r_l(n, k) \psi \left[\frac{\sigma_l + \sum_{k=1}^{K_s-1} \sum_{v \in A_s} a_l(v, k, k) u'_{cl-1}(n+v, k)}{\sigma_l + \frac{r_l(n, k)}{1+r_l(n, k)} b_l(k) u'_{sv}(n)} \right] \quad (3)$$

[Fukushima]

Fukushima restricts the value of \tilde{U}'_{st} only by requiring that it be greater than zero. The a_l values are learned as the network self-organizes using a set of training inputs [Fukushima and Miyake]. There is no requirement that they be normalized. In fact, $\tilde{U}'_{st}(n, k)$ is allowed to take on any value greater than 0. As a result, the cells respond best to the highest values in their input window. The limitation of the neocognitron is that because of this unnormalized correlation calculation, the highest-valued inputs will always dominate the network and the neocognitron is only able to recognize stick figures, such as the letters and numbers with which Fukushima has successfully tested it. The neocognitron has been demonstrated successfully recognizing grey-level images of

airplanes; however, in this case the network based its recognition on the bright spines of the aircraft. I tested this by inserting the spine of one aircraft into the body of another. The aircraft were consistently recognized as being the type whose spine they possessed.

The structure of the neocognitron, which allows for the moderate displacement of features within a level, is useful and important. It reflects the manner in which we perceive ourselves as being able to function. We can recognize and mentally reconstruct objects even when they are deformed, and we are able to recognize the deformed pieces of the object for what they are, in their deformed positions. We do not construct for ourselves a pristine object. Neither could a vision system hope to be able to build an accurate model of what it recognizes if it could not recognize a deformed object and then extract information about that object from where the components lie. What is needed is to extend the neocognitron within this structure so that it can deal with more than simple stick figures.

One way in which to extend the neocognitron is to replace the non-normalized correlation function. By replacing the calculations for the U_i cells with a normalized correlation function, the network can be used to recognize and reconstruct grey scale patterns. The revised equation is:

$$\tilde{U}_{il}^t(n, k) = r/(n, k) \psi \left[\sigma_c \frac{\sum_{k=1}^{K_{l-1}} \sum_{v \in A_l} u_{cl-1}^t(n+v, k) a_l(v, \kappa, k)}{\sqrt{\sum_{k=1}^{K_{l-1}} \sum_{v \in A_l} u_{cl-1}^t(n+v, k)^2 \sum_{k=1}^{K_{l-1}} \sum_{v \in A_l} a_l(v, \kappa, k)^2}} + \sigma_w \frac{\sum_n U_{cl+1}(n) W_{cl+1}(n)}{\sqrt{\sum_n U_{cl+1}(n)^2 \sum_n W_{cl+1}(n)^2}} \right] \quad (4)$$

I call the network resulting from this modification a pseudo-neocognitron, because it maintains the structure, but not all of the equations of the neocognitron. The a_l in this equation can be learned by the network; however, in this particular implementation, I decided to use a fixed function. The key to choosing the function was to select one which could be used to reconstruct grey-scale patterns. Daugman, Mallet and others have used wavelets to reconstruct images [Daugman 88; Mallet]. They have shown that the fidelity of the reconstruction is dependent on

the number of wavelets used. However, good-quality reconstructions can be created with a relatively small set. Daugman used a set of Gabor filters for his wavelets.

A pseudo-neocognitron which uses a limited number of Gabor filters for the a_i can provide a limited reconstruction of a square input for recognition (Figure 33). If the square is modified slightly, the same network is also able to provide a partial reconstruction of the modified square, not the pristine image. The reconstructions are limited by the number of Gabor filters used in the network. This partially successful reconstruction seems to indicate the possibility of developing a robust classifier/reconstructor using the structure of the neocognitron. The pseudo-neocognitron will require further modifications to its learning processes to adapt it to the use of Gabor filters, but these also offer the possibility of building a more extensible system based on the generality of the mappings of the Gabor filters. A network with a_i constructed from these filters is not limited to simple junctions and line-endings as was the cognitron, nor is it limited to the specifically-learned features produced by the neocognitron's learning.

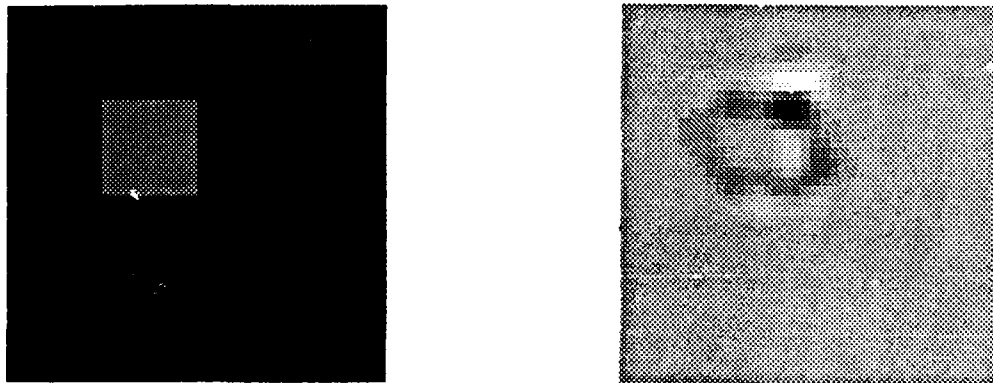


Figure 33: Pseudo-Neocognitron Input and Output

3.6. Attention Mechanisms in Visual Systems

3.6.1. The Superior Colliculus

One problem which is of great interest is how to direct attention in a visual system. In the primate visual system it is known that one of the many systems involved in this process is the superior colliculus. The superior colliculus is located in the posterior aspect of the midbrain section of the brain stem (Figure 34). It is divided into seven layers.

The upper three, or superficial, layers of the superior colliculus receive almost all of their inputs from the visual system. These inputs are received directly from the optic nerve prior to their passing through the lateral geniculate body to which these layers themselves have outputs. The neurons of the superficial colliculus also receive inputs from area 18 of the visual cortex. They have projections to the pulvinar in the anterior section of the thalamus from which they continue their projection to the visual and surrounding areas. The pulvinar also receives inputs from the visual areas. The cells of the superficial layers are mapped to the contralateral visual field of the retina, with a disproportionate amount of area devoted to the center of the field [Diamond; Sparks and Jay].

The deeper layers receive inputs from a wider variety of areas. They have afferent connections from visual, auditory, tactile and motor areas of the cortex. Efferents from the deeper layers project both upward into a wide variety of cortical areas and to structures involved in eye movement as well as downward to areas of the brain stem involved in motor control. Surprisingly, there is no evidence of any strong connection between the superficial and deep layers of the superior colliculus.

Experimentation involving primates has shown that the sensory-related cells of the deep layers have receptive fields which are co-respondent with differing types of sensor stimulation. That is, the field remains roughly constant regardless of whether the innervating stimulant is

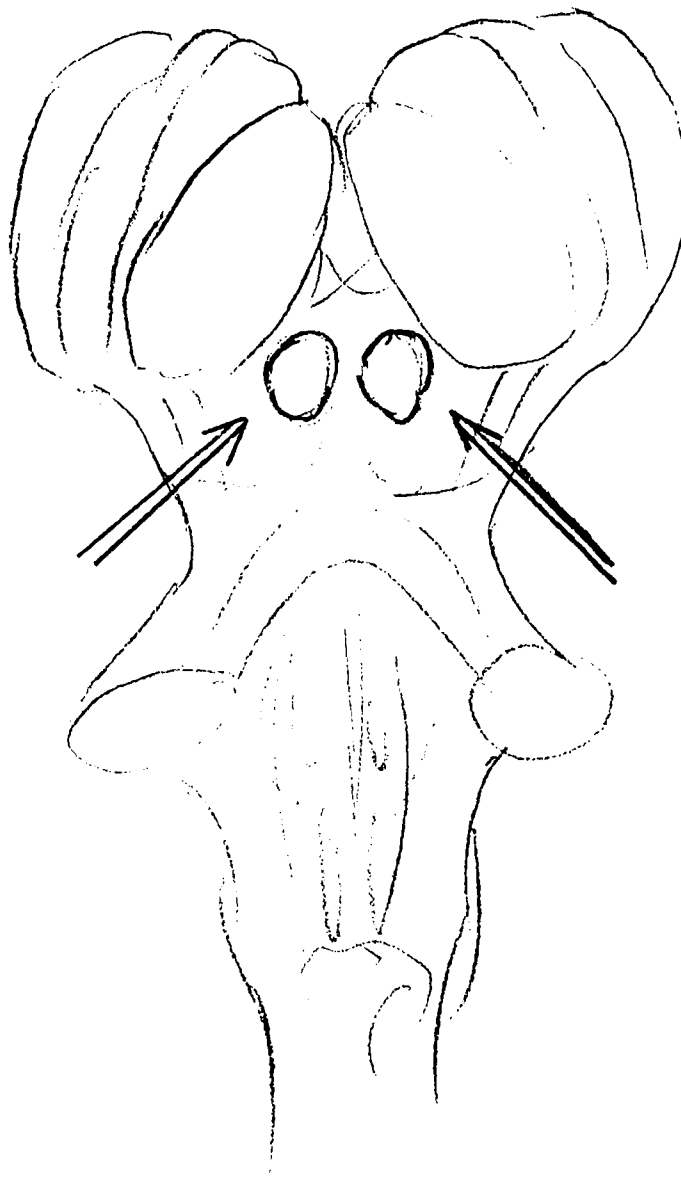


Figure 34: Superior Colliculus in the Mid Section of the Brain Stem

auditory or visual. These fields are mapped not to the retinal field, but rather to a motor-error coordinate field. This mapping reflects the difference between current eye position and desired eye position. These cells are responsive only when a stimulant in their receptive field is accompanied by an eye movement toward that stimulant. For this reason it is believed that these cells

do not regulate attention, but respond to it [Sparks and Jay]. Other studies have confirmed that the response of the cells of the superior colliculus is consistent with the initiation of eye movement [Wurtz].

The apparent lack of connections between the superficial and deep layers of the superior colliculus suggests that the original purpose of the structure has undergone changes. In cats and other intermediate vertebrates there are more direct connections which are heavily involved in attention mechanisms. In the lower (non-cortication) vertebral nervous systems, reptilian and teleost, etc., all of the optic nerves terminate in the optic lobes (tectum), the functional equivalent of the colliculus. At that evolutionary stage there is a direct connection between the stimulus and attention; in fact almost every stimulus is attended to, so there is no need for advanced mechanisms to determine which are meaningful. If a bug-type stimulus presents itself the frog flips out its tongue and intercepts it. If a large shadow falls across its path the frog dives to the nearest patch of blue (presumably water). However, adaptation and evolution appear to have imposed further processing requirements, which have pushed increasing amounts of the processing formerly accomplished in this region into the cortex [Polyak, pp. 306-308]. Therefore, to find the attention centers we need to look into this pathway.

3.6.2. The Posterior Parietal Cortex

The posterior parietal cortex receives afferent connections from the pulvinar, as well as from the other visual areas. The neurons in this area have localized visual receptive fields. In primates these cells respond selectively when a stimulant in their receptive fields is attended to. This response is the same whether the attention is accompanied by an eye movement towards the stimulant, or whether the stimulant is attended to in some other manner. This tends to indicate that the posterior parietal cortex is involved in the attention pathway prior to any initiation of eye movement [Wurtz]. In humans tested using Event-Related Potential (ERP) measures, enhanced

responses in the parietal cortex were noted when attention was paid to a stimulus on the contralateral side [Harter and Aine].

The indications of the data acquired by Wurtz in his experimentation are also supported by clinical observations. People who have damage in the posterior parietal areas tend to be unable to direct attention to objects in their contralateral visual fields. They experience deficiencies in spatial orientation, eye movements, and pointing. The effect is not a complete loss of visual ability in the field. They are able to identify items if directed to them, but are unable to discern their location and are generally reluctant to acknowledge the presence of these items [Goldstein; Wurtz; Holtzman et al.].

It is reasonable to assume that the posterior parietal areas serve some purpose and don't act as merely a relay of attention information. One plausible model is that the posterior parietal areas combine information from the visual areas about features present in a particular scene with information about what is considered significant for attention from the frontal cortex and other areas. These are all areas from which afferent pathways have been identified. This is consistent with the known localization of the discrimination/identification function to the temporal lobe [Goldstein]. For both areas, visual field data are combined with moderating frontal data to provide a result - attention in the posterior parietal cortex; discrimination in the inferior temporal lobe.

Based on this model of the posterior parietal cortex function, there are two problems involving attention mechanisms which need to be explored. The first of these is to find a mechanism in the visual pathways which can provide attentional indicators - that is, a list of areas which indicate features where attention can potentially be fixed. The second problem is to find a source to distinguish among the acceptable attentional indicators and focus on a particular attention event. This source supplies the search strategy and attentional goals. Our focus will be on exploring Gabor filters as a possible solution to the first problem - attentional indicators - although we will also briefly explore the second.

3.6.3. The Gabor Transform as an Attention Mechanism

The Gabor Transform has already been identified as a model for mechanisms found in the cells of the visual cortex. The Gabor transform has the advantage of responding in a spatially, frequency-localized manner. The correlation of a Gabor envelope with an event in a field reaches a peak when the size and orientation of the event is nearest that of the components of the wavelet. Over- and under-sized objects as well as oblique objects cause a decrease in the correlation peak. Careful selection of a wavelet set can provide responses on key features of a scene, such as edges, corners, distinctively sized features, etc. The sensitivity of the Gabor filters to edges is notable as "people fixate on contours much more frequently than they fixate on homogeneous areas of a picture [Gould 76, p. 326]."

One of the more important areas for human survival is the ability to interact socially. For this, the ability to read and exploit facial expressions is a highly critical skill. Any explanation considered for human visual systems must be able to account for its responsiveness to facial features. Figure 35 shows a trace of eye movement when looking at a face. The high concentration of fixations on the facial features is notable. Yarbus also noted the tendency of subjects to dwell on facial features, even in photos of a lion and a gorilla [Yarbus]. It should also be expected that plausible explanations for visual attention mechanisms would be attuned to the human form. This would be important for continuation of the species. Expectations of this type are completely reasonable if one recognizes the extent to which the visual systems of other species are devoted to sexual attraction. In fact, the visual system of the horseshoe crab appears to serve the sole purpose of locating potential mates [Barlow].

A set of images can be created by correlating an image with Gabor envelopes with rotations of 0, 30, 60, and 90 degrees. These images can be combined by taking the most extreme value of the four correlations at each pixel location and putting it into a new image. The result (Figure 36) indicates that Gabor filters can be tuned to have a high response profile for facial features.

Further tests have indicated that the same Gabor filters when correlated with an image of a more distant perspective respond to both the male and female form. At the same time, this set of filters will respond to a wide variety of features which appear to be important to visual attention (bright regions, doors and windows on buildings, etc). Thus Gabor filters as an attention mechanism meet an important biological test.

The use of Gabor transforms as attention indicators also works well with the two-stage model of perception [Chapter 1; Rabbit]. In this model people first rapidly recognize an object as a whole and then identify the components of that object, or they build the object from some clustering of component objects. The clusters people use are dependent on situational specifics, such as the task being performed, or characteristics of the scene, but they do have some general properties in common. The clusters tend to be high-information areas, are grouped as sets of identical symbols, and have similar dimension. The same set of Gabor filters which could be used for high level recognition could provide clustering for directing attention to build the object from detailed clues.

In this mode Gabor filters can be combined with some type of serial search strategy with which to focus attention. One such strategy would be to focus attention first to the most highly responsive areas of a scene. Another would be to search the scene in some specified order (top to bottom, bottom to top, center to sides) and focus the attention on any points which respond above a threshold level. Yet a third possibility would be to skip from one region of dense concentration of attentive indicators to another.

A search method based on the density of attentional indicators could be the result of an inhibitive center-on surround network with a winner-take-all model. In this case the highest-density attentive indicators would cause one of the cells in that area to win out over those of other areas. The attentive act could then be used to provide an additional measure of inhibition to the recently attended area. This would allow other concentrations to give enough strength to one of their

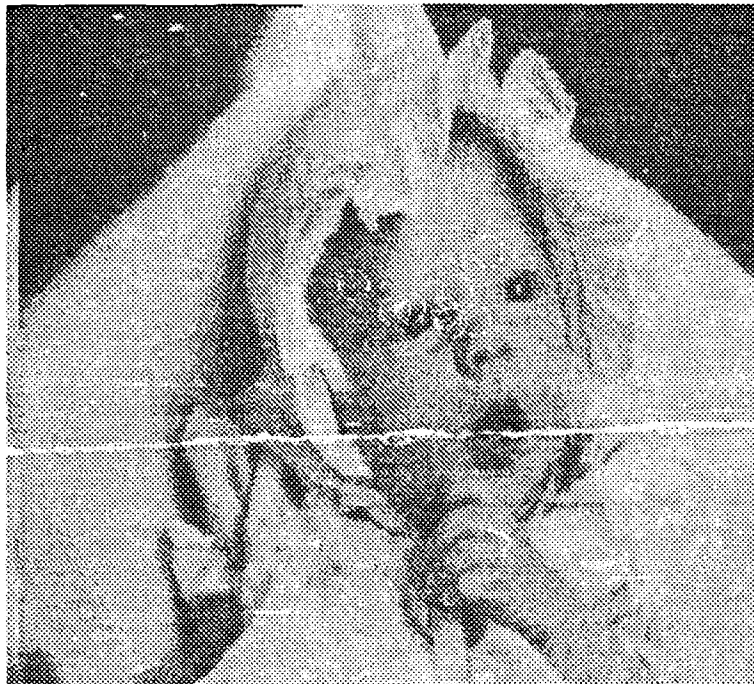


Figure 35: Typical Eye Movements When Viewing a Face



Figure 36: Correlation of Gabor Filters with an Image of a Face

members to allow it to be chosen next. As the inhibition from attending the most dense area wears thin over time, attention could return to that area. This type of a network would allow each area of potential attention to be responded to in proportion to the density of its attentional indicators. Support for focusing the search in this manner can be found in the cells of the posterior parietal cortex itself. "The response of these cells may be attenuated or disappear altogether if the same stimulus is used in eye movement or when the animal is not fixating [Treisman et al. p. 312]." A similar model has been explored for explaining the behavior of frogs and toads in selecting prey from multiple possible targets, and for determining a pathway to get to prey confined behind a barrier [Arbib and House]. A "max-picker" network model which could possibly be adapted to perform this task has been proposed by Suter [Suter and Kabrisky].

Another possible mode in which Gabor filters could be used as an attention model would be in comparing a known or model scene to an actual scene. Images correlated with Gabor filters can expose second and third order differences in scenes where statistics such as averages, standard deviations and modes provide no useful differentiation [Turner]. An example is shown with a Farside Cartoon. In the original, a couch snake has blended himself into his environment by adopting a statistically similar camouflage (See Appendix A, Figure 2). However, by using Gabor filters oriented at 0 and 90 degrees (Figure 37), the snake can easily be exposed as not belonging in the model scene. This attention mode would provide useful survival and hunting mechanisms for the user. It would allow the hunter to pick out prey even when it has attempted to adopt a disguise, and would allow the possessor to avoid potentially dangerous situations, such as predatory couch snakes. The fact that humans are able to make such discriminations is what makes possible the humor of the Farside cartoon. If people were unable have their attentions aroused by the obvious attempt at camouflage by the snake, the drawing would not inspire humor, but rather a difficult search to identify and locate the snake.

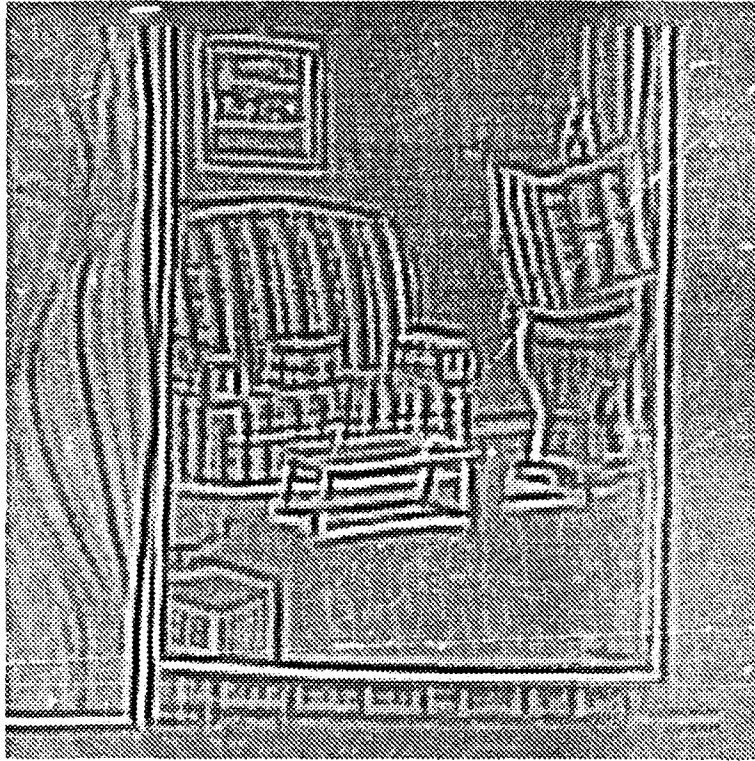


Figure 37: Combined Correlations of 0 and 90 Degree Gabor Filters with the Couch Snake Image

Of the many possible search strategies it is normal for humans to use a wide variety of them, adapting their strategy to the occasion [Snyder and Taylor; Yabus; Rabbitt]. Yabus demonstrated this by showing how the eye-movement patterns of people changed when the motivation for searching a scene changed [Yabus]. In one instance he would ask a test subject to identify the material circumstances of the people in a scene. This produced a distinctive eye movement trace. A totally different trace was produced if the same subject was asked about the emotional state of people in the scene (Figure 38).

In addition to the motivation for searching a scene, search strategies can be affected by the types of information which are available in a scene. When colors are present, test-subjects will generally begin their search for a specified object by scanning those items of identical colors. Only afterwards, or if the choices cannot be identified by color, will they attempt further

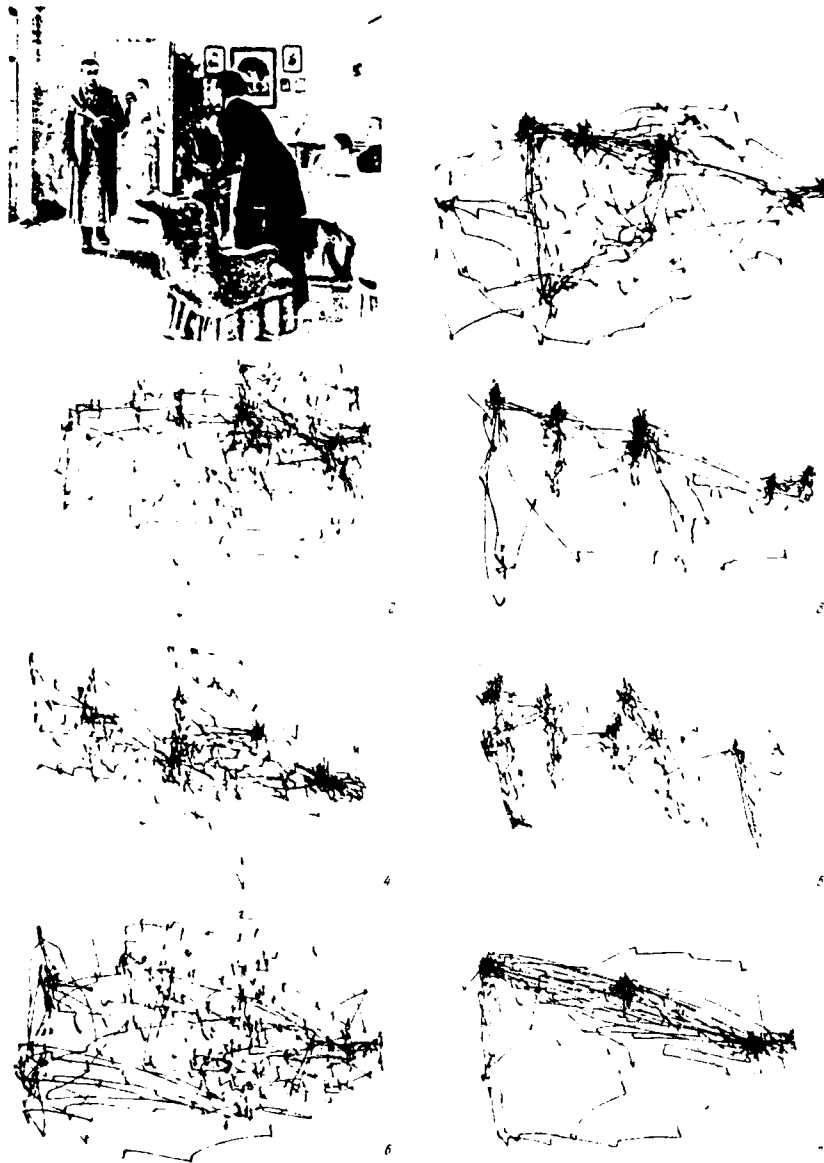


Figure 38: Eye Movements Elicited by Different Verbal Prompts [Yarbus, p. 174]

differentiation based on other factors. Tests of search strategy have focused primarily on the location of artificial structures and differentiation by geometry, size and orientation. They have, in general, not looked at natural images in an attempt to see which components have been the object of various searches. This is the result of two difficulties. First, they have not had a reliable base-attentional indicator to compare to eye movement traces, and second, there is the problem of separating what is based on the search motivation and what is a function of the attentional

indicator. If it can be shown that there is a common attentional indicator regardless of search motivation, then the strategy is the method used to select which of the attentional indicators was most appropriate. Otherwise, the attentional indicators themselves must be considered to be a function of search strategy.

Knowing human search strategies could provide important clues as to how an automated device can conduct its own search of an area. As people generally do better than machines in pattern recognition tasks, they can be assumed to be the experts, and to have developed what may not be the optimal search strategy, but is at least a very effective choice. Knowing how search strategies are altered by intentions and desires can also provide lessons on how an automated system can adapt itself to changing requirements. Finally, knowing in advance the search strategy and attentional indicators which will be used by the operators will allow designers to create optimized control displays. In these displays the most important instrumentation can be designed to receive the maximal amount of attention, while those of lesser importance can become less obtrusive. This could potentially increase the effectiveness of the operator and at the same time, lower the learning curve required to effectively operate the controls.

3.6.4. Experimental Design

One possible basis for an attentional indicator is the Gabor filter. Gabor filters can be used to handle a wide range of phenomena such as color, motion, intensity, spatial frequency and shape. The following experimental design will test the theory that Gabor filters can serve as base-attentional mechanisms and to try to identify some search strategies using these. The experiment should provide sufficient data to give some indication of search strategies both in the case of general natural scenes, and in the specific case of identifying features on videomicrographs of VLSI circuits. The main focus in this investigation will be on attentional indicators, rather than search strategies.

For this experiment, mainly physically existing objects (to include VLSI circuits) are used as they do not provide an artificial base attention indicator such as: alphanumeric characters [Robinson et al.; Gould 69; Prinz; Mocharnuk]; geometric figures [Snyder and Taylor; Gould 73]; disks [Ford et al.]. The use of natural objects also serves to force the subject to look directly at the objects to identify them. In scenes with a limited set of test objects, visual discriminations can be made when stimuli are presented near the periphery [Findlay and Crawford]. This effect is further avoided by increased density of objects found in natural objects. Such crowding serves to limit the size of the usable visual field [Mackworth]. Testing using primarily natural objects also allows the human visual system to work in the environment to which it has been tuned by evolution.

To control the types of base-attentional indicators used, the images shown to test subjects are grey scale still video. This eliminates strategies based on color, movement and binocular disparity. Each subject is shown a variety of images, each of which is accompanied by a text designed to engage different search strategies. After data have been collected on the eye movements of each subject, the scan patterns are compared to a Gabor transform of the same scene. If Gabor filters can model a base attention indicators in humans, it would be expected that most saccades will be to areas near those which also show strong correlation peaks on the Gabor transformed image. It can be expected that there will not be a complete convergence as it has been noted that the eye fixations do not always center exactly on target locations; in fact, they can often be as much as four degrees off-center [Snyder and Taylor]. Further problems can be expected from overshoots and intermediate steps in eye movements [Robinson et al.]. The distance from a Gabor peak can also be expected to be influenced by the characteristics and accuracies of the equipment. Therefore, a reasonable area must be covered in searching for a peak to correlate with each fixation.

3.6.5. Experimental Setup

Data collection for the oculometer experiments has been conducted at the Helmet Mounted Oculometer Facility (HMOF) of the Armstrong Medical Research Laboratory (AMRL). The HMOF has the capability of providing accurate determination of eye gaze angle with respect to the helmet, and the helmet position. From these, the position of eye fixations can be calculated for any eye gaze surface.

The oculometer itself consists of a miniature charge coupled device (CCD) camera mounted on a helmet. A halogen lamp is also mounted on the helmet. The light from the halogen lamp is filtered to allow only the near infrared (IR) components to pass through a collimator and be reflected from a patch of reflective coating on the helmet visor into the eye. Some of the light is reflected from the cornea. Another portion enters the pupil of the eye. A part of this is reflected from the retinal surface. The reflections are picked up in the CCD camera and used for tracking the eye movements. When the eye is focused in the central portion of the eye field the eye direction can be calculated by comparing the center of the light reflected from the cornea to the light reflected from the pupil (Figure 39). At the extremes of eye movement there is no longer any reflection from the pupil so the angle of the eye is determined from the shape of the reflection from the cornea.

...

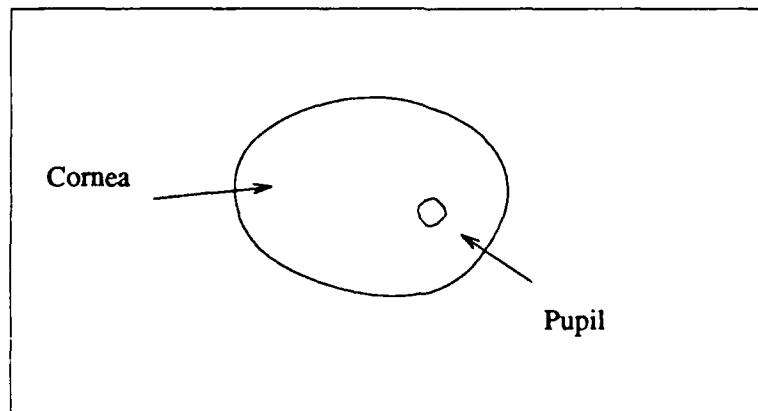


Figure 39: Corneal and Pupal Reflections as Seen by CCD Camera

The helmet, with the added encumbrances of the oculometer equipment, weights just 3 pounds, 13 ounces. This is only slightly greater than the weights of standard Air Force helmets in use today: 3 pounds, 4 ounces and 3 pounds, 8 ounces. This should be little enough to provide only a marginal influence on the subject's head and eye movements. The IR light source produces 228 microwatt/cm² [Mallory] which is well under established safety limits [HMOF]. The IR lighting has no effect on eye movements.

The helmet also contains a Honeywell magnetic helmet-mounted sight (HMS). The HMS provides the position of the subject's head, which is integrated with the data from the oculometer to determine the direction of the eye gaze from a nominal location.

The data from the oculometer and the HMS are collected into a Data General Eclipse computer, type S/130. The oculometer light source and video are connected through a Data General Nova computer. The Data General Eclipse uses the data to compute eye line-of-sight with respect to a fixed coordinate system. The results are passed to a Digital Equipment Corporation MicroVAX II computer which uses the line-of-sight data to compute where the eye gaze is directed. These data can be stored to a file on demand.

When calculating the eye gaze the computer uses a linearization model unique for each subject. The purpose of the linearization technique is to account for the deformities and differences in each individual's eyes. The linearization model is constructed by having the subject look at fixed points on a known linearization grid while the helmet is held in a fixed position. Because the head is held in a fixed position all gaze data result from eye movements. The collection of linearization data is done at a special wall mounted board with lights at known positions. The data are gathered into individual models.

Images were presented to subjects using a Silicon Graphics Iris Workstation. The images are stored on disk in bitmap format. The console of the Iris is displayed on a screen in front of the subject using a projection television. The display of images is controlled from a terminal

attached to one of the serial ports of the workstation. The other serial port is connected to the computer controlling the collection of data from the oculometer. By using double buffering on the Iris it is possible to get presentation times on the order of one frame rate. This is done by maintaining two images in memory. The first of these has two colors: a dark background with a white fixation point in the center. The second maps to a 256 level grey scale. While a picture is being loaded into the background buffer, the fixation point is maintained on the screen. When ready for presentation, the buffers are swapped and the scene instantly (in one frame scan) appears in grey scale. A signal is sent to begin data collection just prior to swapping the buffers.

The helmet also has an audio system which can be used to give instructions to the subject, and to pick up any responses. This system is used to record verbal prompts and the subject's responses.

3.6.6. Experimental Procedures

Subjects for this experiment were recruited from the VLSI design sequence at AFIT. This selection is made because the students are familiar with VLSI structures. These students can be expected to be a fairly representative sample in other respects. Each subject receives an eye examination prior to entry into the experiment. This detects any abnormalities in the subject's vision. Participant requirements are given in Appendix E. A copy of the release form required from each subject is also given in Appendix E.

After the subjects have been through the normalization procedure they are prepared for the main test. For testing the subjects are seated in the mock cockpit wearing the helmet containing the test apparatus. Here, their heads are not constrained. In leaving the head free, the only unnatural constraint on the subject is the minor inconvenience of wearing the helmet. This encourages naturalness in the eye movements of the subjects. Once seated the subject undergoes a short calibration procedure.

The first scenes subjects are shown is a set of nine screens with fixation points. These scenes are used to boresight the data collection by insuring that the center of the screen is read as the center of the subject's eye field. The scenes are also used to normalize the data by providing the extreme coordinates to which the collected data will be mapped. The upper left screen position is mapped to coordinate (0,0) and the lower right to (512, 512). The calibration screens also provide a means of determining the accuracy of the normalization data and of the mapping of measured fixations into pixel locations. This is done by comparing the final four calibration-point results to predetermined coordinates for where those points should be. When the final test points are not measured within a reasonable accuracy the boresight procedure can be rerun, or, in extreme cases, the normalization data can be changed to a generic model, or retaken.

After calibration the subjects are presented with 50 images. The images are presented in random order. When not presenting an image the viewing screen remains blank, except for the fixation point. Prior to each image's being presented the subject is given a verbal cue intended to elicit a specific behavioral pattern in the subject's eye-movements. Several of the images have more than one possible verbal stimulus. After the stimulus has been read the subject is presented with the picture for a duration of about 60 seconds. The exact time is controlled by the tester. During this time the subjects are expected to complete any tasks required by the verbal cue. At the end of the period the screen is switched to the fixation point and the next image prepared. An audio record is maintained of the verbal cues and the subjects' responses. Eye position recordings are maintained for the time period of exposure of each image.

3.6.7. Data Analysis

After the eye-movement traces have been acquired they are analyzed to determine the fixation points. The oculometer only measures the location of the eye every 1/60th of a second. It does not determine which points are fixations and which are only intermediate points on a sac-

cade. A simple method would be to determine each point where the movement out of the point is in a different direction than the movement into that point. All eye movements are direct linear transits, although a saccade may consist of more than one movement with a pause, too short to collect data, between movements. These multiple movements would confuse a system which merely looks for changes in movement direction. The solution then appears to be to require a number of consecutive measurements with the same position recorded. This too would be inadequate, as the eye does not stay completely stable during a fixation but rather tends to jitter and wander (Figure 40).

The method used to extract these fixations and saccades from oculometer results can create large differences in the way data are interpreted [Karsh and Breitenbach; Widdel]. Too liberal a construction of fixation points can result in excessive numbers of points being found either as a result of involuntary eye movement during a fixation or as a result of slow movement or directional changes during a saccade. On the other hand, too stringent requirements will force the loss

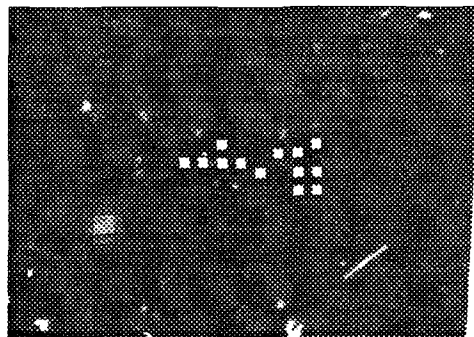


Figure 40: Example of Expected Eye Jitter During a Single Fixation

of valid fixation points. For the purposes of this study the required balance is that an attempt be made to limit the loss of fixation locations, but that the grouping of multiple fixations in a localized area is allowable.

The data recording eye-movements contain four types of records. The first of these is a normal eye location calculated using the corneal and pupil reflections. The second is an eye location calculated using an elliptical algorithm which proved to be inaccurate. The third record-type includes eye blinks and other tracking and recording failures. The fourth and final record-type is a travel record. These are recorded when the blur induced in the video image indicates the eye gaze is moving. To determine a fixation point the eye-movement data are processed until two successive points which lie within a degree of each other in the visual field are found. These points indicate a possible beginning of a fixation. The following points are then checked for jumps of greater than a degree of the visual field. If none is found within 160ms the centroid of the points is labeled as a fixation point. 160ms as a minimum duration was selected as being near one generally-accepted minimum of 180ms required for a fixation [Widdel], although other, shorter times are also accepted [Widdel; Karsh and Breitenbach]. Points for which data were not available were accepted as falling within the time constraints. Termination of a fixation is determined either by a movement greater than one degree from the prior point in the fixation, a movement greater than two degrees from the second point in the fixation, or a recorded movement in the data. The validity of this method for determining fixations was confirmed by overlaying fixation points, and a two-degree diameter circle about them, on a scatter plot of the eye-movement points (Figure 41).

Once the fixation points have been located, they are checked to see how well they can be fitted to a Gabor envelope. Because the fixation points may not be centered exactly on the position to which they were attracted, 9 points scattered about a two degree foveal field are considered (Figure 42). These points are then correlated with a set of Gabor filters. Each set

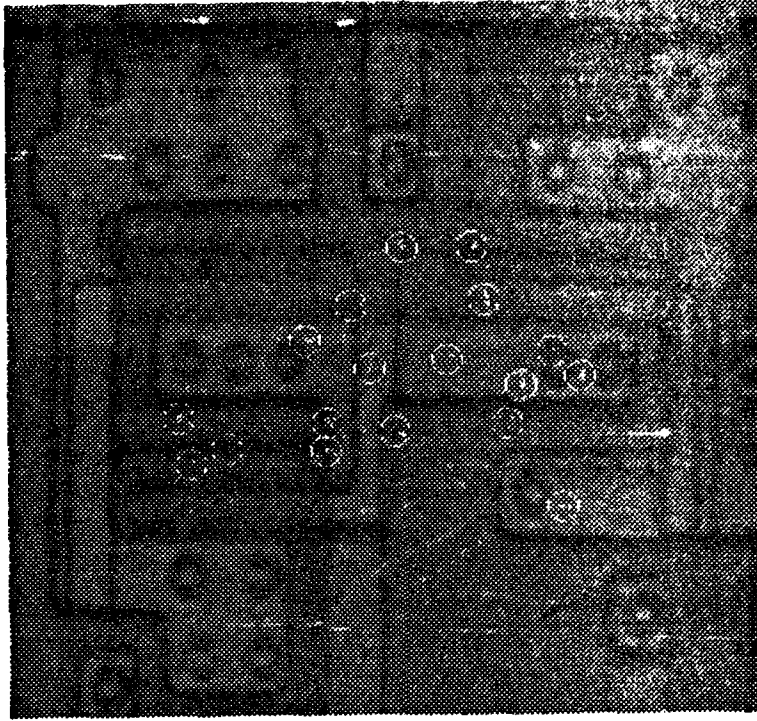


Figure 41: Fixation Circles Plotted on Eye Movement Points

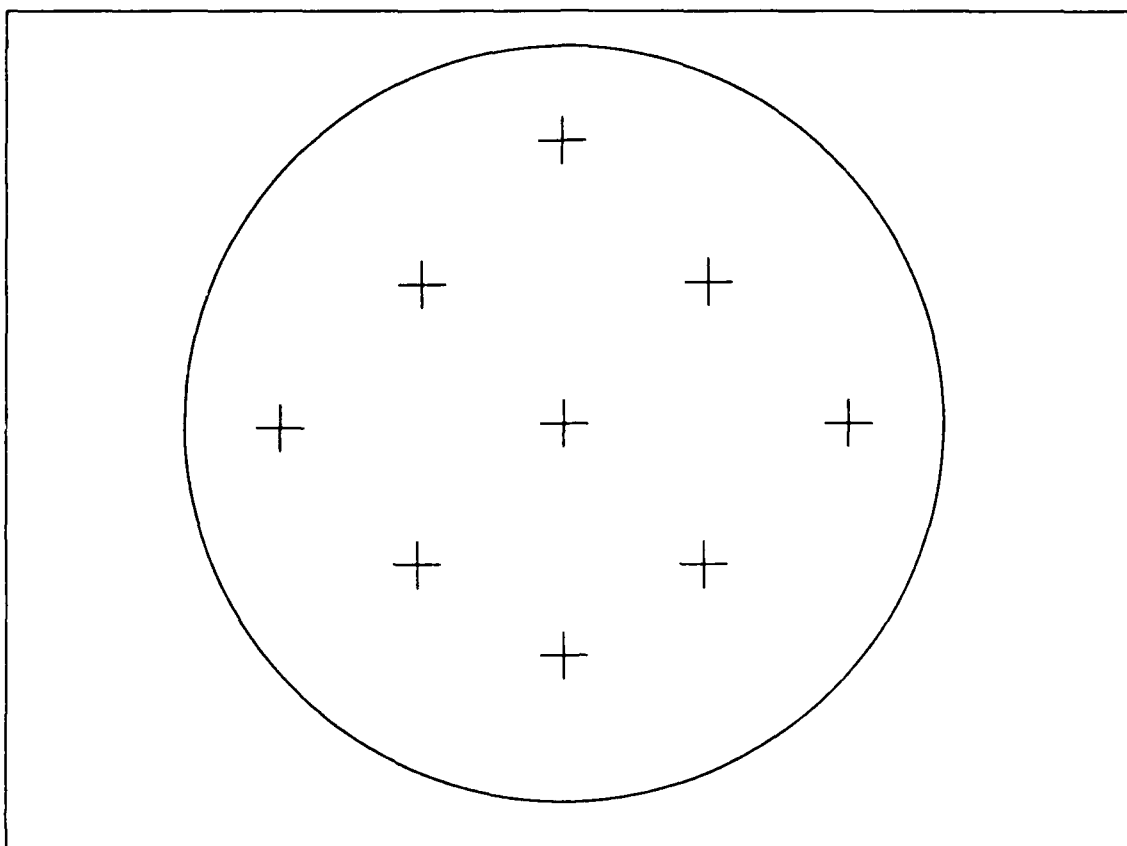


Figure 42: Distribution of Test Points within the Foveal Field

includes 7 sizes, 6 orientations, 3 wave types and 3 spatial frequencies for a total of 378 correlations per point (Table 1, Figure 43). The sizes range from just under two, to just under seven degrees of the visual field (18 to 66 pixels). The orientations were chosen to be representative of those found in the mammalian visual system. Likewise, the phases and spatial frequencies were chosen to represent possibly wide ranges of such values in biological vision systems. For each fixation point the maximum correlation peak is chosen and the filter parameters are recorded. In order to somewhat reduce the computational load, once a correlation peak above 0.5 has been discovered, no further points within the fixation circle are assessed.

To provide a basis of comparison 100 additional points are chosen at random from the input image. At each of these points the maximum correlation of the set of Gabor filters is selected.

Table 1: Gabor Filter Characteristics for Fixation Analysis

sizes	orientations	wave types	spatial frequencies
18	0	$\text{sine}(\omega)$	1
26	30	$\text{sine}(\omega + .5\pi)$	2
34	60	$\text{sine}(\omega + \pi)$	3
42	90		
50	120		
58	150		
66	180		

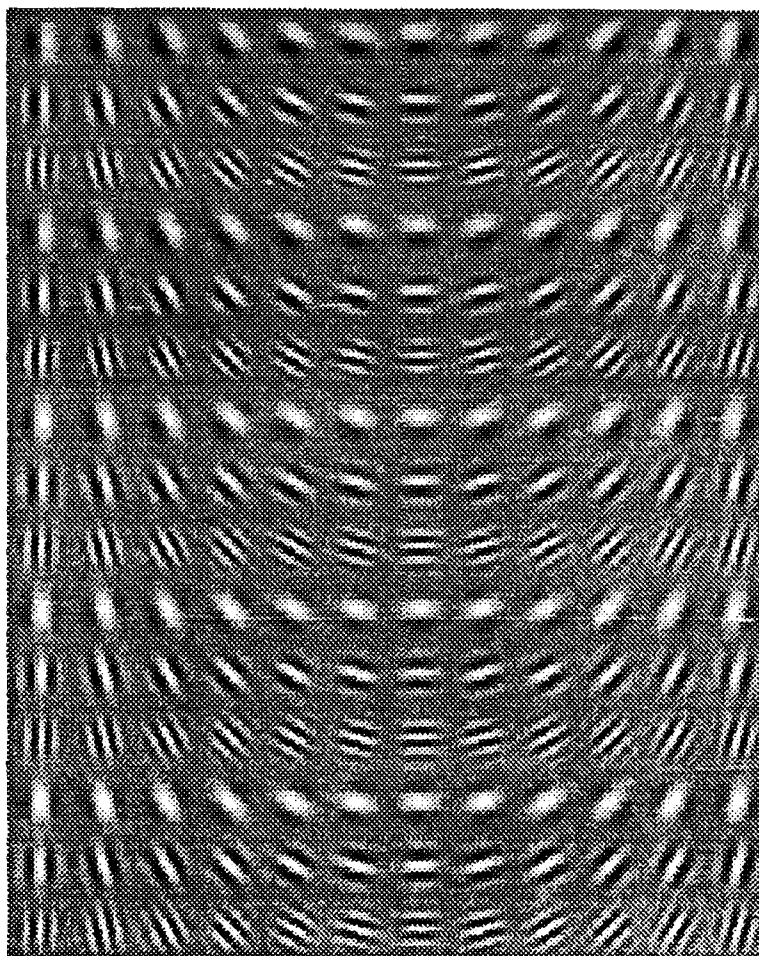


Figure 43: Sample of Gabor Filters - Size 18

With these values and the fixation point values, a null hypothesis is established to test whether the mean of the maximum Gabor correlations of the population of random points is less than the mean of the maximum Gabor correlations of the population of fixation points. Since examination of histograms shows the data to appear to be normally, or nearly normally, distributed, a T-test can be used to accept or reject the null hypothesis [Hines and Montgomery; Walpole and Myers]. The T-test also has the advantage of being robust to aberrations in the normality of the data. However, as there is no certainty as to the equivalency of the variances of the populations, the sufficiency of the sample size cannot be determined [Hines and Montgomery].

Table 2 shows the results of the T-test on samples from six subjects. Not all subjects have results for all images. In some cases equipment problems (overheating, etc.) prevented the gathering of data. In other cases the oculometer system was unable to properly track the eye (blinks, tears, dirty contacts, etc.). In 89% of the samples, the null hypothesis was rejected for a 95% confidence interval. The rejection of the null hypothesis forces acceptance of the alternative hypothesis that the correlations with Gabor filters for fixation points exceed those of random positions on the images. The differences in the populations can also be illustrated by constructing histograms of the relative frequencies of the various Gabor correlation values (Figure 44). The increased correlation values for fixation points imply that Gabor functions can accurately model the basic attentional mechanism for human vision, and in fact may, as they are known to be present in the appropriate areas of the cortex, prove to be the only mechanism. This mechanism is then further modified by some higher-level search function.

Table 2: Results of T-Tests

Subject:	A	B	C	D	E	F	Subject:	A	B	C	D	E	F
image	X = Reject H_0						O = Fail to Reject H_0						
1	X	X	X	X	X	X	26	X	X	X	X	X	X
2	X	X	X	X	X	X	27	X	-	X	X	X	-
3	X	X	X	X	-	X	28	-	X	X	X	X	X
4	X	X	X	X	X	X	29	X	X	X	X	X	X
5	X	X	X	X	O	X	30	X	O	X	X	-	X
6	X	X	X	X	X	X	31	O	X	X	X	X	O
7	X	X	X	-	X	X	32	O	O	X	X	O	O
8	X	X	X	X	X	X	33	X	X	X	X	X	X
9	X	X	X	X	X	-	34	X	X	X	X	X	X
10	X	X	X	X	X	X	35	O	X	X	X	X	-
11	X	X	X	X	X	X	36	O	O	X	O	O	O
12	X	X	X	X	X	X	37	X	X	X	X	X	-
13	X	X	X	X	X	X	38	O	X	X	X	-	X
14	X	-	X	X	-	X	39	-	X	X	X	X	X
15	-	X	X	X	X	X	40	X	X	X	X	X	X
16	O	O	O	O	-	O	41	X	X	X	X	-	X
17	X	X	X	X	X	-	42	X	X	X	X	X	X
18	X	X	X	X	X	X	43	X	X	X	X	X	X
19	X	X	X	X	X	X	44	X	X	X	X	X	X
20	-	X	-	X	-	X	45	X	X	X	-	X	X
21	X	X	X	X	X	X	46	X	X	X	X	-	X
22	X	X	X	X	X	X	47	O	O	X	O	O	O
23	X	X	X	X	X	X							
24	X	X	X	-	X	X							
25	X	X	X	X	-	X							
Subject:	A	B	C	D	E	F	Total						
Reject H_0	35	40	46	41	33	37	232						
Pictures	43	46	46	44	37	45	261						89%

It is instructive to look at the images where the test failed to reject the null hypothesis. Of the 24 cases in 8 images where this occurred five images had multiple failures. One of these was a scene with a variety of white boxes (Figure 45). The subjects were asked to fixate briefly on each box. In viewing the fixation points it was seen that some subjects only cast their eyes in the direction of each box instead of insuring that they fixated directly on the box. This left the fixation point in the empty space between boxes. The fixations which were on actual boxes

A) .

```

      **
      **
      **
    ** **
  ** ** **
** ** ** ** ** ** ** ** 

```

Mean = 0.372783, Variance = 0.005429.

B) .

```

      **
      ** **
      ** **
    ** ** **
  ** ** ** ** 
** ** ** ** 

```

Mean = 0.450063, Variance = 0.011104.

C) .

```

      **
      ** **
      ** **
    ** ** **
    ** ** **

```

Mean = 0.413352, Variance = 0.004952.

D) .

```

      **
      **
      **
    ** **
  ** ** 
** ** 
** ** ** 

```

Mean = 0.456469, Variance = 0.014074.

E) .

```

      **
      **
      **
    ** **
  ** ** **
** ** ** ** 

```

Mean = 0.453246, Variance = 0.006379.

F) .

```

      **
      **
      **
      **
      **
      **
    ** **
    ** ** 

```

Mean = 0.450126, Variance = 0.001497.

G) .

```

      **
      **
    ** **
  ** ** ** 
** ** ** ** 

```

Mean = 0.431711, Variance = 0.013330.

Figure 44: Histograms of the Relative Frequency for Gabor Correlation Values for a Representative Image: A - Random Points. B-G - Subjects

tended to lie along the edges of the boxes. As a result, the histograms for these images show that although there were more points above the mean for the baseline, the fixation points with correlations above the mean tended to group at higher values (Figure 46). Another interesting observation from this image is that subjects appeared to follow a rather random ordering in fixating on

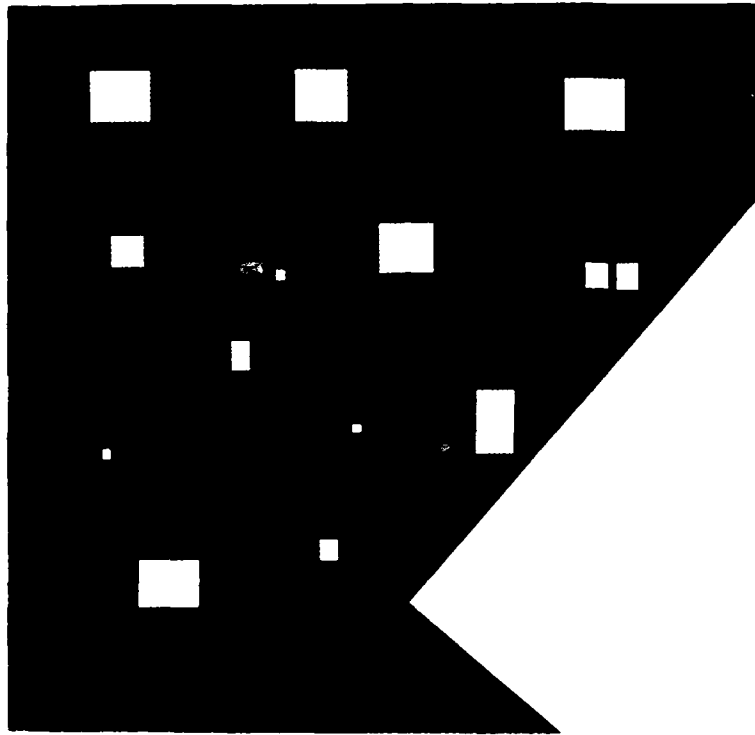


Figure 45: Box Image

the boxes rather than working from left to right, or top to bottom as might be expected since we like to think of ourselves as rational ordered creatures.

Two other pictures had 5 failures to reject the null hypothesis. In one of these the subjects were instructed to find the transistors, a feature which in this scene had a lot less texture and edges than did the image as a whole. Interestingly, for all five subjects who viewed this image the mean of the population of Gabor correlations with the fixation points was statistically lower than the mean of the correlations with random points on the image with a 99% significance level. This is suggestive that in certain cases the absence of attentional indicators may be used in conjunction with a search technique. This was also indicated, although to a lesser extent, in another image in which the subjects were asked to determine the type of diffusion in some transistors. The area in which the diffusion was located did not have as high a Gabor filter response as did the

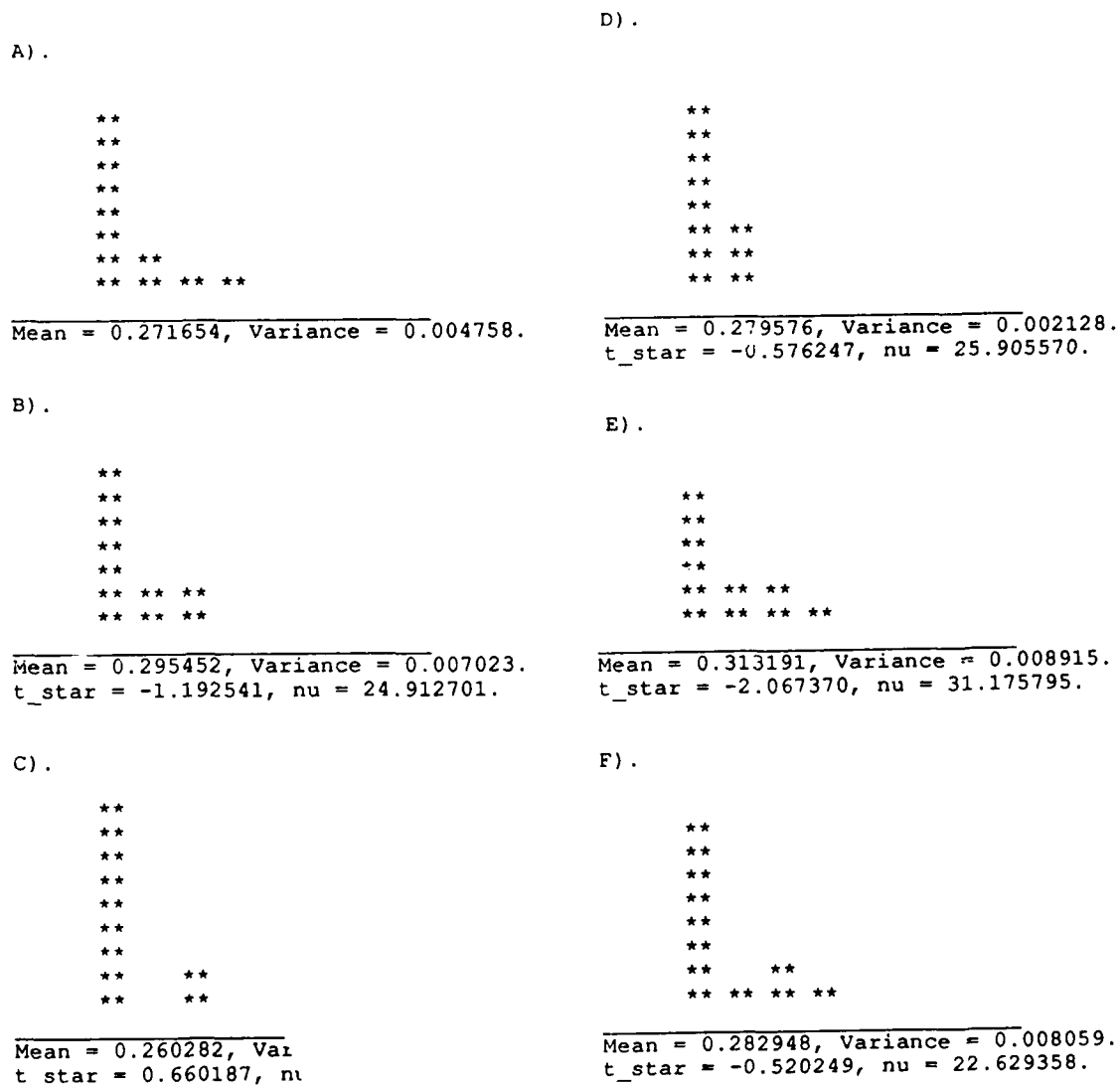


Figure 46: Histograms of Correlation Distributions for Box Image

other areas on the scene. Part of the process of determining the transistor type is to look at the surrounding structures as well as at the actual diffusion patch. Here the means of the fixation populations were only slightly lower than those of the population as a whole; this would be the result of a combination of observing both the diffusion areas and the surrounding structures.

There is no such clear indicator of the cause of the failure to reject the null hypothesis in the other image with five failures. For this image subjects were asked to trace an electrical path from one corner to another on an image of a VLSI circuit. No clear pattern of groupings among the means of the fixation populations was observed, nor was there any indication found in the ordering of the fixation points.

The final image with which there were multiple failures to reject the null hypothesis had two subjects which failed to do so. In both cases they just missed rejecting the null hypothesis. One would have done so at the 94% level and one at 92%. Two of the other three failures to reject the null hypothesis were also near-misses; one would have been rejected at 74%, one at 90%. These images both had small numbers of fixation points and it might be expected that had this not been so the null hypotheses would have been rejected at the appropriate levels. No clear explanation exists for the final image in which the fixation mean approached that of the total population.

The next logical step after isolating the fundamental attentional indicators is to determine how these are used in conjunction with a search mechanism. A couple of simple possibilities recommend themselves. These include searching first the strongest peaks of correlation with Gabor functions; searching the nearest Gabor peaks; or some combination of these strategies. A few simple tests of the data disproved these possibilities. There is no correlation between the ordering of the fixation points and the strength of correlation with a Gabor function, nor is there any with distance either from the first fixation point, nor with any other. In fact the only obvious trend is a fairly consistent decrease in the product of the inverse of the distance from the first fixation point and the square of the strength of the correlation, or

$$est = \frac{1}{dist} * peak^2 \quad (5)$$

during the first 400-800 msec of the presentation of an image (Table 3). This could suggest a search strategy which is determined in advance and changed when sufficient information about

the scene becomes available, or it may be an artifact of the test methodology. In view of the consistency with which it occurs it appears to be the former. One other possible search strategy is apparent from observing the position of the fixation points - the order of fixations appears to be partially dependent on the density of Gabor peaks in the region of the fixation.

3.7. Light, Color and Lasers

Light is an interesting phenomenon, a simple definition of which is not totally possible. For now we will define it as the portion of the electro-magnetic spectrum which mammals use in their visual systems and the immediately adjacent spectral areas. This includes the range of 400-700 nanometers (nm) of color, the infrared (IR) spectrum up to 12,000 nm, and the ultra-violet range down to 100 nm. It is in these ranges of the spectrum that interaction with and reflectances from materials is most effective in providing information about the properties of the materials. Nature in choosing this range for her vision systems to operate selected the most responsive region of the curve. This is the area where effects due to macroscopic or grouped structures and microscopic, or individual structures most overlap. Longer wavelengths interact mainly with macro-structures, shorter wavelengths with micro-structures. The area of maximal span of interactions can be portrayed as in Figure 47. Typical of microscopic interactions is the interaction of short wavelengths with the individual particles in the atmosphere, which results in the blue appearance of the sky. A macroscopic effect would be the reflection off a crystal surface. A gamma-ray penetrates practically everything, changing its course only when it strikes the nucleus of an atom. Radio-waves

Table 3: Estimators For The First 5 Fixations for Sample Images

Image	Fixation 1	Fixation 2	Fixation 3	Fixation 4	Fixation 5
1	0.005307	0.002079	0.001395	0.003786	0.001717
2	0.004336	0.005262	0.001807	0.003390	0.001288
3	0.002906	0.000608	0.001128	0.000707	0.001685
4	0.002482	0.002262	0.001903	0.000927	0.000893
5	0.002324	0.002347	0.001024	0.000281	0.002799

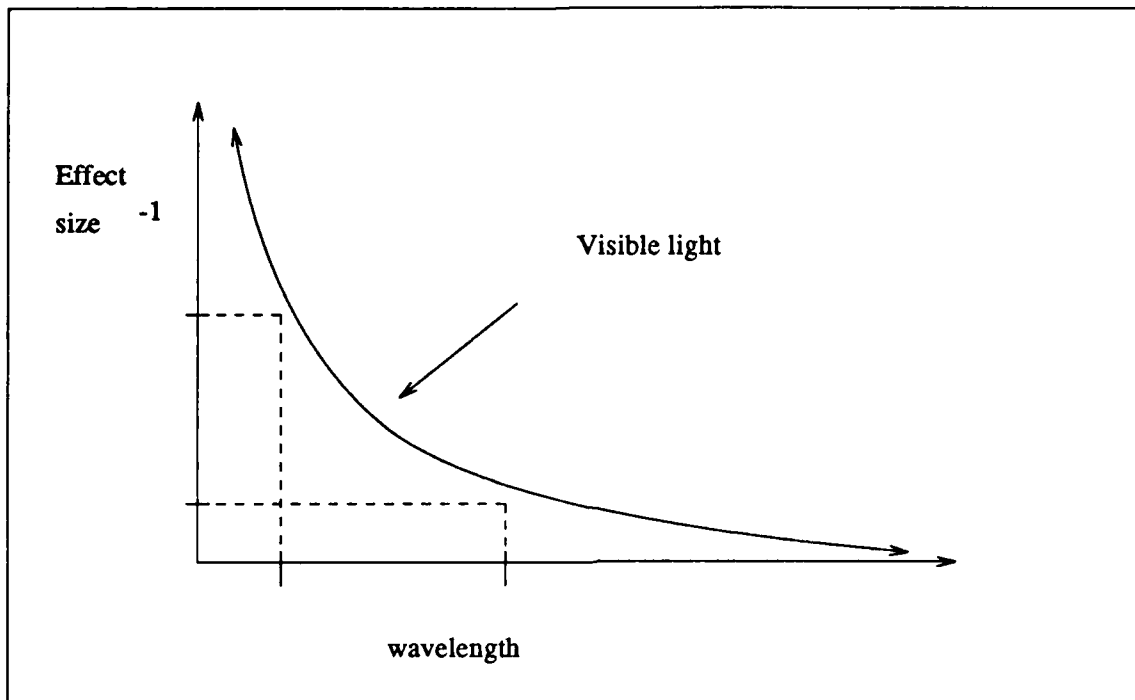


Figure 47: Microscopic vrs. Macroscopic Interactions with the Electro-Magnetic Spectrum

on the other hand are immune to the effects of a single particle and need a larger assemblage, such as an antenna, to detect them, or a large mass, such as a hill, to block them.

In this well-chosen portion of the spectrum, there are three methods nature uses which take advantage of spectral information to describe the visual world. The first of these is to measure the intensity or density of light coming into the sensor. The next method is to use a three-color system to provide a better model of the light falling on the sensors. Finally, nature uses an opponent-color system, built upon the other systems, to enable the information gathered by them to be more efficiently put to use.

3.7.1. Monochromaticity

The basic and simplest system used in the eye is the rod cell. These cells are sensitive to a broad spectrum of light. They provide a monochromatic source for the human vision system.

Many animals have never developed a multicolor system, and others, just as with people, have a monochromatic system which they use when the conditions don't allow use of their full visual capabilities. Such conditions might include: reduced lighting, the extremes of the visual field, and quick responses. Much information can be gathered from a monochromatic input. The majority of the light impinging on an eye or a sensor is reflected light; that is, rather than coming directly from some source, the light reflects off some material before entering the sensor. Thus, the information transmitted by the optic nerve is a function of the source, the transmission path, materials off which the light reflects and the response characteristics of the sensors (Figure 48). By parameterizing these factors we can predict the appearance of a particular scene. In theory all that is needed is a set of equations. In practice these equations quickly become extremely complex and unwieldy. There are, however, special cases in which the problem can be somewhat simplified. The spectral distribution of a single source can be measured. If the source is enclosed and focused in a single direction the effects of reflections from materials other than the material of interest are reduced. Maintaining a normal incidence of the light on the surface of a material minimizes the complexity of the equations needed to predict the amount of light returned from the surface, and finally, reducing the length of the path reduces the transmission-path effects.

Sources of illumination can include "black body" radiation, in which the light radiated is dependent entirely on the temperature of the material, or radiation which is dependent on the chemical properties of some gas or other material. A fluorescent light is a good example of the latter, where the light radiated depends on the internal gases and the fluorescence properties of the phosphor coating inside the tube. Candles and matches are representative of "black body" radiators, and, in general, incandescent bulbs can be represented as black body radiators at a temperature somewhat lower than the actual filament temperature [Evans]. The Sun at or near its surface is a representative "black body" radiator, but a great deal of filtering of sunlight occurs before it

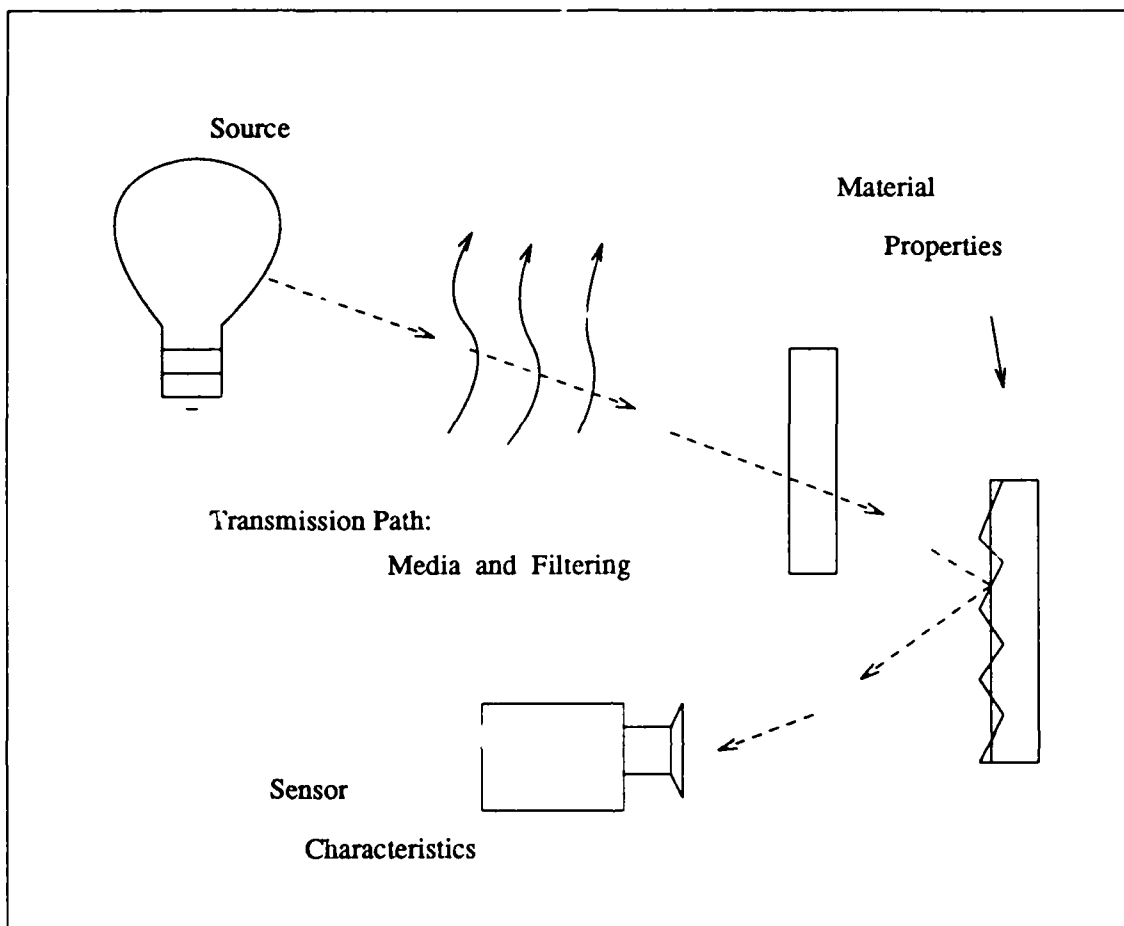


Figure 48: Factors Influencing Transmitted Light

reaches the earth's surface. Filtering can also be used to alter other illumination sources, creating new illuminants with characteristics not otherwise available.

The Commission Internationale de L'Eclairage (CIE) has established a number of standard illumination sources. These serve as references to establish the conditions under which a particular color is observed in a material. The sources include illuminant A - an incandescent light with temperature of 2856 K; illuminant B - representing the noon Sun with a nominal temperature of 4874 K; and illuminant C - average daylight on an overcast day with a nominal temperature of 6774 K. Illuminants B and C can be obtained from illuminant A and the appropriate filters. The

CIE has also established a number of other more specialized illuminants [Chamberlin and Chamberlin]. When these established illuminants are not available, it is possible to plot the spectrum of a particular illuminant through the use of a spectrum analyzer.

The amount and type of radiation returned from an object is a function of the four basic interactions between lightwaves and materials. These include reflection, refraction, absorption and emission. Each of these is wavelength-dependent. Reflection occurs when light strikes an optically opaque surface. The angle of reflection will be equal to the angle of incidence. The material may be reflective over the entire spectrum of incident light, or only over some portion. In those portions of the spectrum in which the material is not wholly reflective, the light entering the sample will be refracted. This will occur in proportion to the difference in the permeabilities, or refractive indices, of the materials. For a light wave traveling from one medium into another, if the refractive index of the second medium is higher than that of the first, the light will bend toward a line drawn normal to the boundary of the materials. If the second medium has a lower index than the first, the light will bend away from the normal. The amount of bending is wavelength-dependent within each material. It is possible and normal for light to pass through several material interfaces, each adding its own bend to the light. It is also possible for light to pass through several boundaries, be reflected, and then pass through the boundaries again. The returned radiation is also subject to constructive and destructive interference.

Absorption occurs when the energy in the impinging waves is sufficient to change the atomic state of the material. In the Bohr model of the atom, the nucleus is surrounded by a number of electrons. These electrons lie in bands at discrete distances from the nucleus. There is a given number of electrons which may occupy any particular band. The inner bands are always filled with their allotted number of electrons, which, barring unnatural acts by prying physicists, remain in place. In the outer bands, however, the electrons are free to move from band to band within certain constraints. Energy is absorbed as electrons in the material change the band in

which they lie. Each change of band requires a quantum of energy equal to the difference between the energy levels of the bands the electrons occupy. This quantum can be expressed as $E_2 - E_1 = h \cdot f$, where E_1 and E_2 are the energy levels, h is Planck's constant ($6.62 \cdot 10^{-27} \text{ erg-sec}$) and f is a frequency of vibration. Photons representing these frequencies, and only these frequencies, will provide the correct amounts of energy for the electrons to make the band jumps [Hallmark and Horn]. Thus absorption is dependent upon the particular frequency of the impinging light, as well as on the physical characteristics of the material. Emission is the spontaneous release of photon energy as the electrons return to lower energy states.

In the case of thin films, reflectance, refraction and absorption combine to create a highly frequency dependent mixture. If normally incident light is used, the light reflected from the surface can be described by equation 6:

$$R = \frac{\rho_1^2 + \rho_2^2 - 2\rho_1\rho_2\cos(\delta)}{1 + \rho_1^2\rho_2^2 - 2\rho_1\rho_2\cos(\delta)} \quad (6)$$

where

$$\rho_1 = \frac{n_1 - n_0}{n_1 + n_0}$$

$$\rho_2 = \frac{n_1 - n_2}{n_1 + n_2}$$

and

$$\delta = \frac{4\pi n_1 d}{\lambda}$$

[Nussbaum, p 186]

n_0 , n_1 and n_2 are the indices of refraction of the air, the thin-film material and the underlying material, respectively, and d is the thickness of the thin film. This equation is exact enough for

most normal cases, especially those where the only concern is the relative intensity of the returning light. A more accurate equation is

$$R = \frac{g_1^2 + g_2^2 + h_2^2 + 2g_1g_2\cos(2\gamma) + 2g_1h_2\sin(2\gamma)}{1 + g_1^2(g_2^2 + h_2^2) + 2g_1g_2\cos(2\gamma) + 2g_1h_2\sin(2\gamma)} \quad (7)$$

where

$$g_1 = \frac{n_0^2 - n_1^2}{(n_0 + n_1)^2}$$

$$g_2 = \frac{n_1^2 - n_2^2 - k_2^2}{(n_1 + n_2)^2 + k_2^2}$$

$$\gamma = \frac{2\pi n_1 d}{\lambda}$$

$$h_2 = \frac{2n_1k_2}{(n_1 + n_2)^2 + k_2^2}$$

[Heavens, pp. 76-77]

and k_2 is the extinction coefficient of the underlying layer.

A prediction of the expected detectable return intensity of light incident normal to a surface can be obtained by integrating the equations modeling the source and material characteristics over the full range of frequencies, i.e.,

$$I(\lambda) = \int_0^{\infty} R(\lambda)L(\lambda)S(\lambda)d\lambda \quad (8)$$

where S is a description of the response characteristics of the sensor, and R is a formula for the reflection. This model allows for differentiation between materials possessing differing characteristics, or, if the thin film is used, differentiation in the depths of the film. Figure 49 shows a sample plot of intensity versus film depth. This figure shows that there are places in the plot where the intensity is the same, or close to the same, for several depths. It can also occur that the expected intensities are the same for two different material types. If two points with a common intensity are of interest, it is possible to alter the equation to enhance distinctions between the

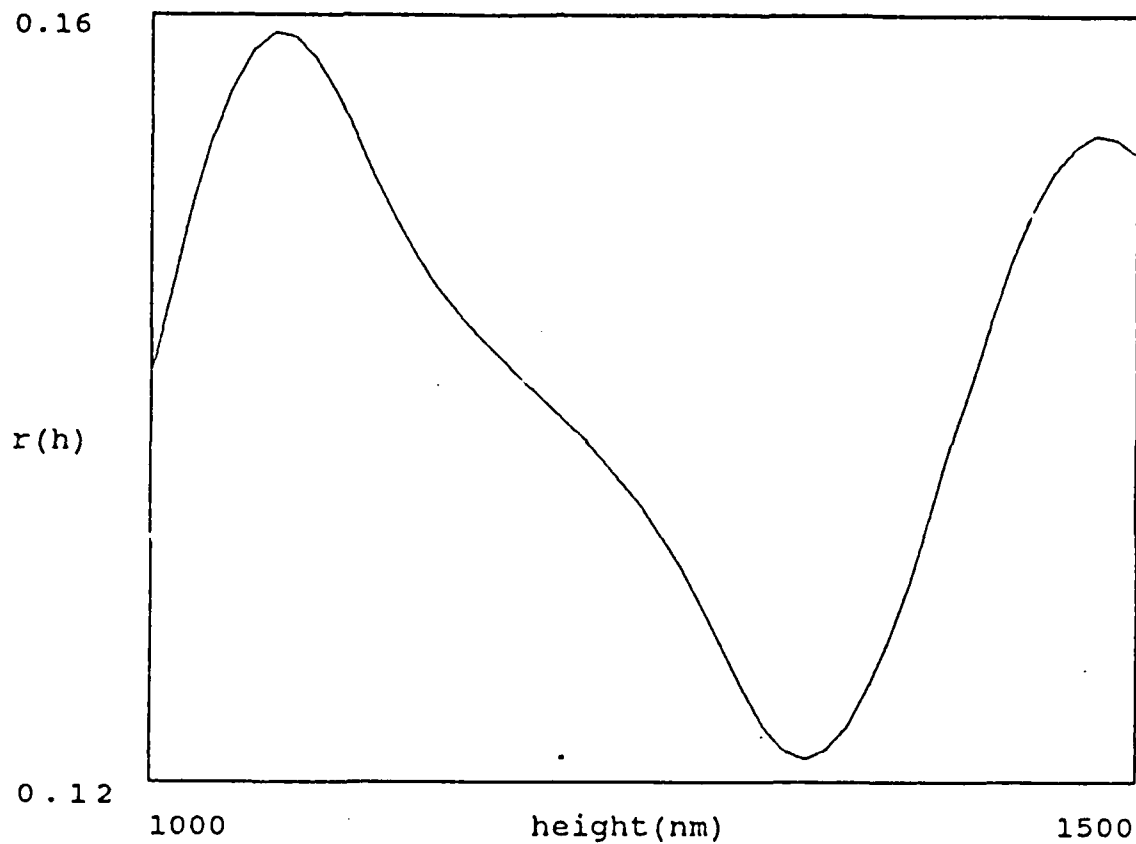


Figure 49: Reflection Intensity Versus Film Thickness

two. This is done by adding a filter, or filters, to the optical system. In this case the equation for the returned intensity is altered to include the characteristic equation of the filter:

$$I(\lambda) = \int_0^{\infty} R(\lambda) F(\lambda) L(\lambda) S(\lambda) d\lambda \quad (9)$$

The filter, $F(\lambda)$, is chosen to allow differentiation between the points. This can be done either heuristically, or by choosing a desired output curve and differentiating the equation to find a solution:

$$F(\lambda) = \frac{I(\lambda) d\lambda + C}{R(\lambda) S(\lambda) L(\lambda)} \quad (10)$$

Often the heuristic selection is easier as there may be only a limited number of filters available, and as the differentiation task can be quite difficult.

This model can also be used to predict the returned intensities of single or limited-frequency illumination, such as the case of lasers, or filtered light sources. In the case of limited-frequency applications, the filter factor is used to model the limitations placed the spectrum, while for single frequencies the need to integrate across the spectrum is eliminated. In either case, the resulting model is dependent upon the material type, or, in the case of thin films, the film thickness. This model could also be expanded to cover non-normal incidence, multiple light sources, etc. Some work has been done characterizing reflectances from materials at known oblique angles of incidence [Augustus; Moss; Wolf and Boulton]. However, the number of considerations in these cases causes the equations to rapidly become unwieldy.

3.7.2. Multi-Colors

People, and in fact most mammals, do not use a simple intensity model in their visual systems; instead they use a multi-color system. These multi-color systems use receptors with different characteristics. It is not uncommon for vertebrates to have a two-color system; most primates use three colors. In these systems the sensors, cone cells in the eye, each absorb a range of frequencies which can be described with some characteristic curve. The ranges for the different types of sensors overlap, but each has a distinct peak response frequency. Such a system does not provide an unambiguous description of the impinging spectrum, but it does reduce the overall data to a more manageable color space.

Humans use a three-color system with peak reception at about 420, 580 and 640 nanometers [Goldstein]. The approximate response curves are given in Figure 50. Because these response curves overlap, and because they cover broad spectral areas, it is possible that two different input signals will appear to be identical. The result is not a unique representation of every possible

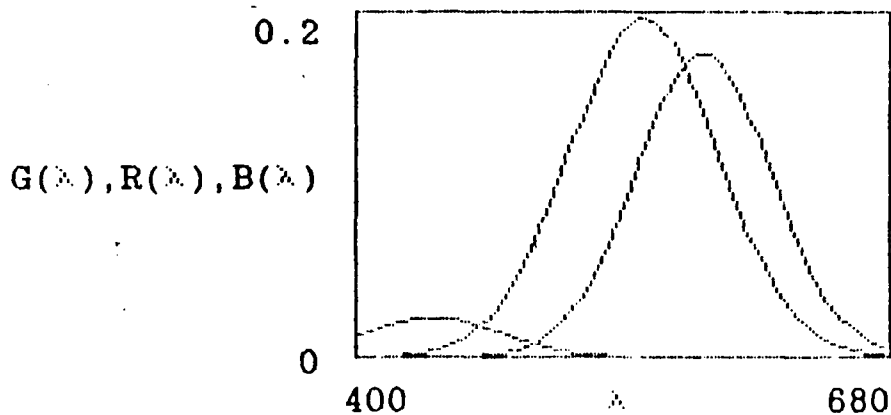
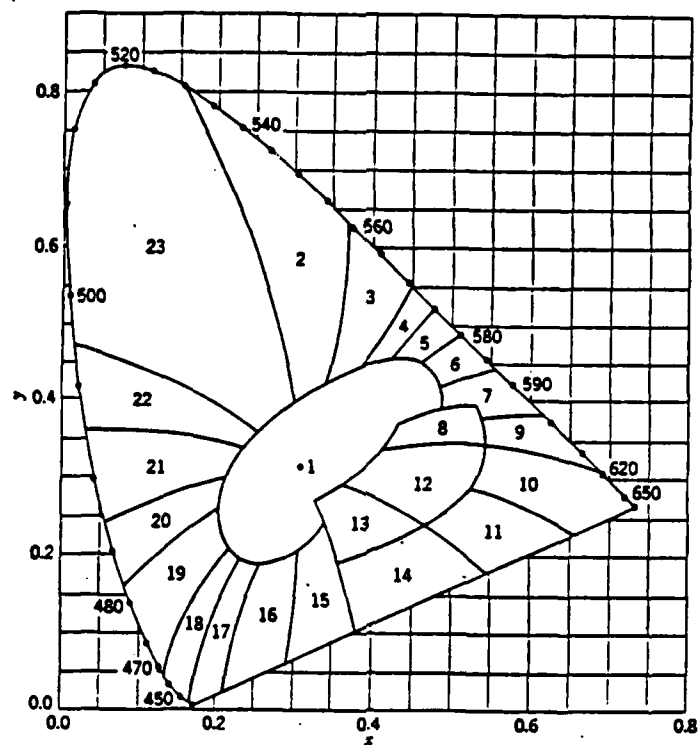


Figure 50: Human Visual Frequency Response Curves

spectral distribution, but an adequate representation which is capable of providing significant spectral information.

There have been many ways devised to represent this three dimensional color space, but perhaps the most common of these is the CIE chromaticity diagram. First established in 1931, with revisions in 1960, 1964 and 1976, this chart maps hue and saturation onto a two-dimensional plot. Luminosity is assumed to be a third dimension directed outward from the surface. On the CIE diagram, colors which are perceptually similar to a human observer appear near one another. Ideally, a plot of colors which appear the same to an observer would constitute a circular area of the chart. This is not always the case; however the 1976 chart comes close. The perimeter of the CIE diagram (Figure 51) consists of fully saturated colors which can be represented with a single frequency. The center of the diagram, $x = .333$, $y = .333$, represents a complete blending of chroma which with luminosity gives a grey scale. The line bending through the center of the diagram represents a graph of black-body radiation versus temperature. The CIE diagram offers us two possible advantages: first it allows a means to plot spectra and predict the



- | | | |
|----------------------|--------------------|-------------------|
| 1. ICI Illuminant C. | 9. Reddish Orange | 17. Bluish Purple |
| 2. Yellowish Green | 10. Red | 18. Purplish Blue |
| 3. Yellow-Green | 11. Purplish Red | 19. Blue |
| 4. Greenish Yellow | 12. Pink | 20. Greenish Blue |
| 5. Yellow | 13. Purplish Pink | 21. Blue-Green |
| 6. Yellowish Orange | 14. Red-Purple | 22. Bluish Green |
| 7. Orange | 15. Reddish Purple | 23. Green |
| 8. Orange-Pink | 16. Purple | |

Figure 51: 1931 CIE Chromaticity Diagram [Augustus, p. 27]

color they represent for a human observer, and second, the diagram provides a means for calculating how different any two spectra will appear to an observer.

Prediction of how a spectrum will appear to an observer is done using equation 10. The equation is used three times. In each case the equation used for $F(l)$ represents one of the response curves for the cone cells of the eye. The three equations give nominal values for X, Y, and Z. The results are then combined and converted to the 1931 CIE diagram axis, where

$$x = \frac{X}{X + Y + Z} \quad (11)$$

$$y = \frac{Y}{X + Y + Z} \quad (12)$$

and

$$z = \frac{Z}{X + Y + Z} \quad (13)$$

where

$$1 = x + y + z \quad (14)$$

In this case, the tristimulus value Y also represents the luminosity. As a result all colors can be represented in terms of x, y and Y [Hardy]. The values of x and y can either be plotted on the 1931 diagram, or they can be converted to the 1976 standard through the equations:

$$u = \frac{4x}{-2x + 12y + 3} \quad (15)$$

$$v = \frac{6y}{-2x + 12y + 3} \quad (16)$$

[Wright, p. 191]

These values can also be calculated directly from the tristimulus values X, Y and Z:

$$L^* = 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 \quad \text{for} \quad \frac{Y}{Y_n} > 0.01 \quad (\text{note}^3) \quad (17)$$

$$u^* = 13L^*(u' - u'_n) \quad (18)$$

$$v^* = 13L^*(v' - v'_n) \quad (19)$$

where

$$u' = \frac{4X}{X + 15Y + 3Z}$$

$$v' = \frac{9Y}{X + 15Y + 3Z}$$

and

³Equations for smaller values of $\frac{Y}{Y_n}$ are given in: [Recommendations].

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n}$$

$$v'_n = \frac{9Y_n}{X_n + 15Y_n + 3Z_n}$$

[Recommendations]

X_n , Y_n , and Z_n represent the tristimulus values of a nominally white object color stimulus. L^* represents the luminosity component of the color. The chromaticity diagram is a plot of u' , v' .

In the case of a thin film, the changes due to film thickness can be plotted as a trace on the diagram. Comparing an observed color to the diagram will then provide an accurate estimate of the thickness of the film. Relevant limitations to using this technique include the variations in perceptual tasks among different people, and the factors within people that affect their perception (expectation, environment, etc.).

The calculated u and v values can also be used to predict the differences between any two colors using the equation:

$$\Delta E_{uv}^* = [(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2]^{\frac{1}{2}} \quad (20)$$

[Recommendations]

The exact threshold where a color difference becomes perceptible depends on the individual observer, as well as the field size, the nature of the surround, the luminance and chromaticity of the surround, and the size of the dividing line between the samples [Recommendations]. Interestingly, the existence of a color difference can be detected before the nature of the change can be determined [Zrenner et al.].

It is possible to develop a similar multi-color system for an RGB camera, or for any other set of sensors, or single sensor using multiple filters. In this case equation 10 is again used with the responses of the color components used for $F(l)$. A diagram constructed using the same methods as those used to construct the CIE diagram will provide information about whether

inputs with differing spectra can be differentiated within the color system. If they cannot, a known trace on the diagram can be used to alter, or construct, a color system which can. For colors which are known to be separable by people, the simplest and most obvious method is to alter the color system to have responses which mimic those of the human visual system. This can be done by working from the tristimulus values for the input spectrum. These values are the result of three systems which each include a source, a reflectance, and a filter which represents one of the tristimulus responses of the eye. The goal is now to replace the single filter with a pair of filters which represent the response of the sensor and a modification that will make the combined response approximate that of the eye in the regions of interest (Figure 52). At times an individual filter can be selected for each input band, but often, as in the case of color cameras, only a single filter can be used with a sensor at a time. This leaves the options of either choosing one filter to cover the entire spectrum, or of using multiple filters in sequence. If the first option is chosen, the filter will have to be some sort of a compromise solution. In this case the filter can be selected by differentiating each of the modified tristimulus filters in turn and solving for the $F(l)$'s. A filter with the best potential compromise for these solutions is selected. Using the filter response characteristics a new trace is plotted on the chromaticity diagram. This can then be inspected to see if the filter provides adequate separation within the color space. If it does not, new compromise filters can be tested until one is found which adequately fulfills the requirements.

When multiple filters are used in sequence, they can be chosen either by differentiating the modified tristimulus equations or through heuristic methods. If filters are chosen heuristically, the system can be checked to determine whether the color-space separation is adequate by performing the appropriate color-space calculations and using a modified version of equation 20 (ΔE). As higher discriminatory levels are required, the system can be implemented as a 2, 3, 4 or higher color-space model.

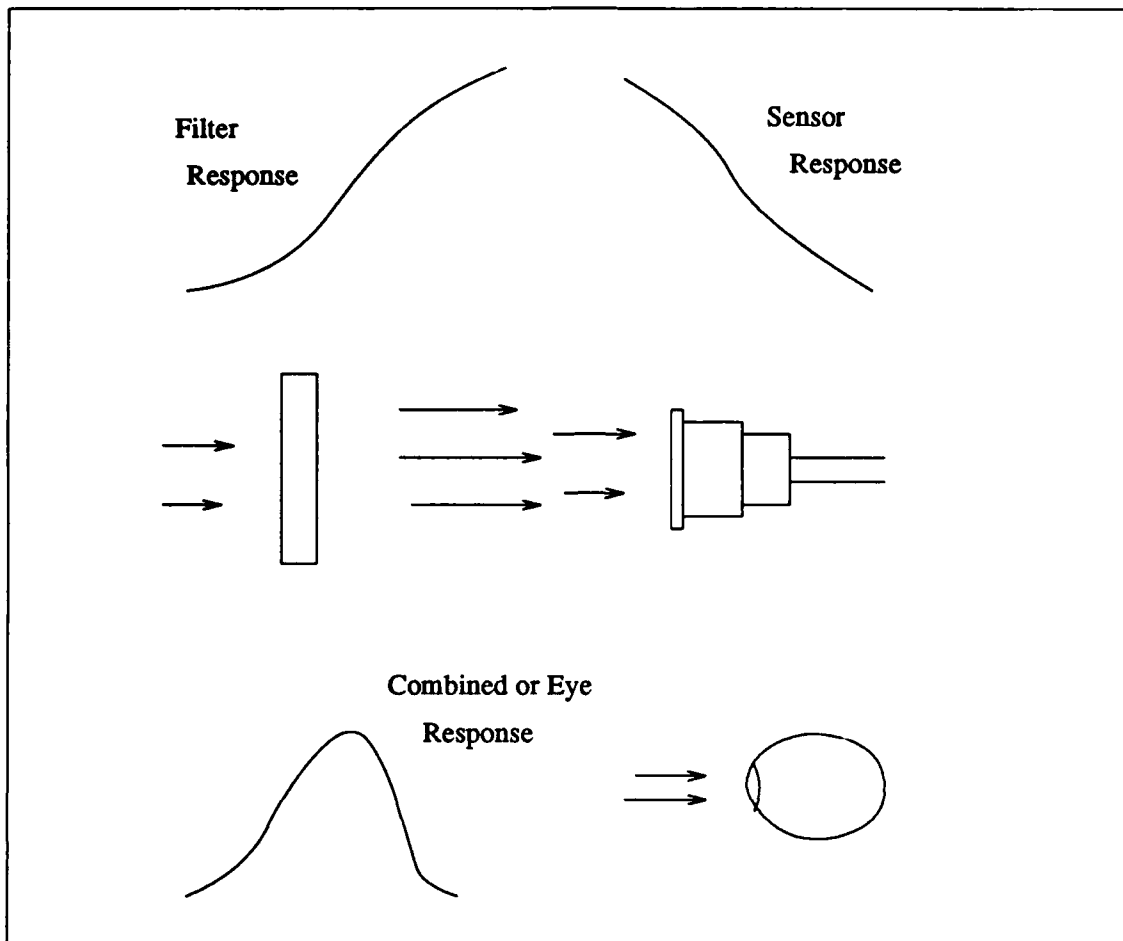


Figure 52: System to Provide Color Response Equivalent to the Eye

3.7.3. Opponent Colors

A multi-color space has a finite area as determined by the characteristics of the sensor. Nonetheless, multi-color spaces represent a great deal more information than can conveniently be dealt with. One method which has been adopted as a partial solution to this problem is to use opponent colors. Opponent color planes, rather than representing a single color intensity, define the differences between two colors. Thus there is more information inherently contained in each plane.

The human uses three sets of opponent colors; primary among these is the light density plane, or dark-light opponency plane. This plane is generally not thought of in terms of an

opponent plane, but doing so proves useful. The other two opponency planes are blue-yellow and red-green (Figure 53). When tracing the path of information received by the eyes, cells which support the opponent color scheme are first found in the retina. Here the inputs of several types of cone cells are combined to produce the mappings. Although greatly simplified, the red-green mapping can be derived from an inhibitory effect on the response of the long-wave, or red, cells by the medium-wave, or green, cells. The blue-yellow response curve is somewhat non-linear in nature, but this can be accounted for by using the difference between the medium-wave and long-wave responses to inhibit the response of the short-wave, or blue, cones [Zrenner et al.]. The actual construction of these functions is done with populations of ganglion cells in the retina which exhibit on-long-wave-center - off-medium-wave-surround, off-long-wave-center - on-medium-wave-surround and other such characteristics [Zrenner; Marr]. These introduce the same lateral inhibitive effects into the opponent color system as are found in the monochromatic portions of the visual system.

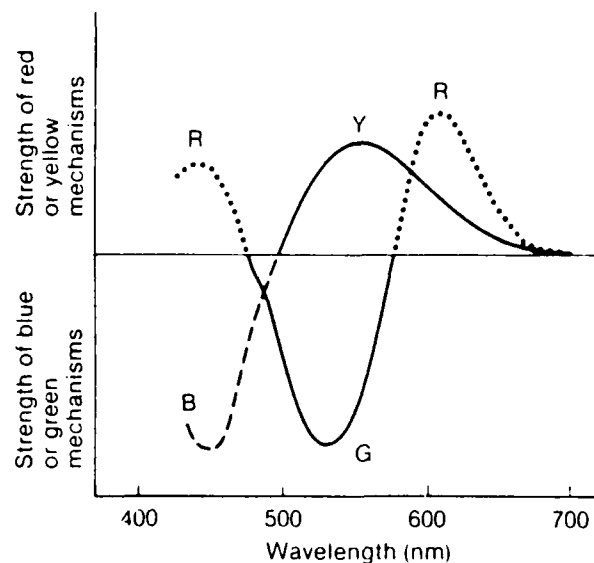


Figure 53: Red-Green and Blue-Yellow Response Curves [Goldstein, p. 125]

From the retina the opponent color projections extend to the lateral geniculate nucleus. Color-sensitive cells are confined to the parvocellular, or tonic, system. From the lateral geniculate nucleus the parvocellular system extends to the visual cortex. Here color-sensitive cells contribute to both color-sensitive areas called blobs (because these areas look like blobs when stained for microscopic examination), and to the orientation sensitive interblob areas. In this manner the color information is encapsulated in several of the functional channels of the brain's architecture. In this manner the color information contributes to edge sensitivity, form, and hue-color determination [Zrenner et al.; Goldstein]. Although useful, the relative unimportance of the actual color is evidenced by both the relatively small number of cells devoted to color in the visual areas, and by the observation that "an illuminant may be up to 93% chromatic, but provided it contains at least 7% 'daylight', surfaces with uniform spectral reflectance - that reflect equally at all wavelengths- will remain achromatic [Marr, p. 250]". Instead, the major significance of the chromatic information appears to come from its contributions to the rest of the parvocellular system.

The opponent color methods can be exploited in a computer vision system. A direct translation from tristimulus values can be made with the equations

$$R+G^- = R(\lambda) - G(\lambda) \quad (21)$$

$$B-Y^+ = -5B(\lambda) + 0.2[G(\lambda) + R(\lambda)] \quad (22)$$

where $R+G^-$ represents the red-green opponent plane, and $B-Y^+$ represents the blue-yellow plane. A plot of these values is given in Figure 54. If this plot is compared with Figure 53, a difference is seen in the RG response at lower frequencies. This is simply a limitation of the equation used to model the response of the red cones. The particular values of the constants in the equation of BY reflect the higher sensitivity of the long and medium-wave cone cells. These equations can be translated to transform the inputs from an RGB camera into an opponent color system. The

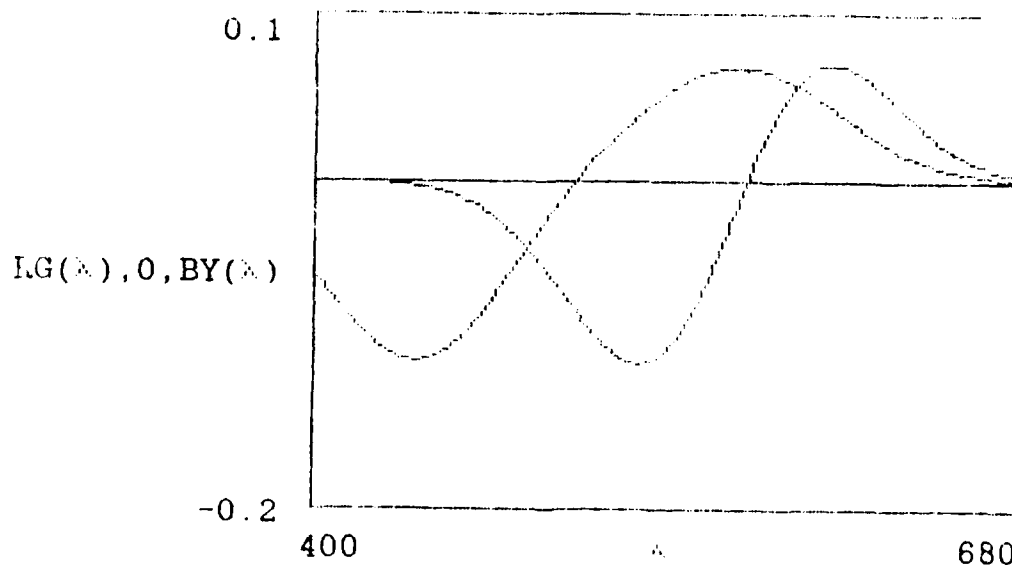


Figure 54: Plot of Opponent Values

constants for $B-Y^+$ are determined by the relative sensitivity of the blue component. When the sensitivity of the blue component approaches that of the other signals, the multiplier on $B(\lambda)$ approaches one and the multiplier of the yellow component $[G(\lambda) + R(\lambda)]$ approaches 0.5. $R(\lambda)$, $G(\lambda)$ and $B(\lambda)$ represent normalized red, green and blue values. The normalization is important as it removes vestiges of the intensity which would otherwise be preserved. Thus an $R+G^-$ value of $(R = 150) - (G = 75)$ would be no more meaningful than one of $(R = 2) - (G = 1)$.

Once the opponent planes have been built they can be used independently or in combination with an intensity plane for a wide variety of tasks. This includes the location of edges, the separation of regions, and the determination of features and their locations.

CHAPTER 4

The AFIT Reverse Engineering System

The purpose of this chapter is to discuss the AFIT Reverse Engineering System (ARES) and the application of the vision system model to ARES. ARES is a product of an effort to reverse-engineer Very Large Scale Integrated (VLSI) circuits using pattern recognition techniques. Reverse engineering is a process whereby the specifications of an object, in this case a VLSI circuit, are derived from study of the actual object. The CAD, Heuristics and Image Processing (CHIP) system provides a research platform upon which ARES is based.

4.1. Background

The reverse engineering project began in response to requirements both within the Department of Defense and from industry to have a capability to obtain circuit specifications directly from a circuit. Often specifications for a particular circuit are not available. This is particularly true in the Department of Defense where the average life cycle for a piece of equipment (15 - 25 years) is much longer than that of the 5 to 7 year life cycle commonly found in the market place. Military logisticians are faced with the problem of finding replacements for failed parts which have not been produced for years. The companies which produced the product may not have kept any specifications for the product, or for that matter, may no longer exist given the highly dynamic situation in the electronics market place.

There are only a limited number of options when a part is no longer available. First, the part can be redesigned to perform the proper function. This is an extremely difficult task for all but the simplest circuits, because, although the normal functions of the device may be well understood, there are often unanticipated functions which the circuit must perform in exceptional cases. Thus in a peacetime, low-threat, low-stress environment, the redesigned part may function

perfectly normally; however, at some unknown critical moment it may fail. Further, redesigning the circuit can be an extremely difficult and complex task if the specifications of the original circuit are not known.

Other options would be to replace the subassembly, or the major end item, in which the component is used. Using the former option is an expensive task which can be subject to the same hazards, though in a somewhat reduced manner, as redesigning only the particular circuit. Engineers can take advantage of the need to redesign a subassembly to upgrade the capabilities of a system, but it is easier to do if the design specifications of the parts of the original subassembly are known. A drawback to subassembly-redesign is that it can be more costly than simply redesigning the failed part. Replacing the entire system, however, is even more costly, and, in addition, can require a great deal more administrative and legislative overhead. This option is even more difficult in times of constrained defense budgets. All of these options can require a great deal of time and money, and all of them, except replacing the entire end item, are most effective if the specifications of the original circuit are known. As an example, in one particular system for which I was responsible, a small digital filter became inoperable. The original replacement cost was approximately \$3. The actual cost of a replacement was more than \$11,000, because the part was no longer commercially available and had to be redesigned and manufactured. The end item sat, unusable, in the motor pool for in excess of nine months awaiting delivery of the part.

Several steps can be taken to ensure that either the part or its specifications remain available. One step which may be taken is to anticipate the termination of the commercial life cycle and to ensure that sufficient quantities of the product are available for future needs. This requires a degree of cooperation on the part of the manufacturers of the circuit, a requirement which is not always easy to fulfill as the manufacturers themselves may not always know that the ultimate destination of their product line, shipped to some intermediate assembler, is the military. Even with

the cooperation of the suppliers, the efforts to maintain a sufficient quantity are fraught with expense and uncertainty. In an attempt to hold down the cost, the logistician may stockpile only a minimally sufficient quantity. Any abnormal demands on this stockpile could wipe out the entire supply. On the other hand, too large a stockpile represents a waste of scarce resources. For this reason, attempts are made, when possible, to find an alternate equivalent source of supply. The Defense Electronics Supply command was forced to pursue these measures in over 100,000 cases in one year alone.

Another step which the Department of Defense has begun to implement is to require that all circuits delivered in DoD equipment come with a complete VHSIC Hardware Description Language (VHDL) specification. This requirement provides a partial solution to the problem for future circuits, but not a complete one. There have been a number of exceptions granted to the policy, and it can be anticipated that there will be more exceptions granted in the future. Further, there is the problem of ensuring that the VHDL description provided does properly specify the characteristics of the circuit it claims to describe. Difficulties in maintaining the quality of the documentation for a product are a common organizational problem. This is so for a number of reasons, among which the least admitted, but most influencing, is that preparing documentation is boring. As a result, when the tasks in a design team are passed out, it is often the most junior, most inexperienced engineer, the recent graduate, who gets assigned the task of preparing the documentation. The senior engineers are used for the "important" design work. As a result the documentation, and the VHDL in particular, may not accurately reflect the actual design of the circuit. Better design techniques, which use the VHDL directly as a design step, may help reduce this type of problem, but are not likely to eliminate it. Even when a VHDL specification is produced which at one stage exactly matches the actual circuit, it is not certain that the two will remain in sync. Changes and corrections made to a circuit in its late stages are less likely to get annotated in the documentation as they may be made in haste in an attempt to get the circuit produced on

time, or because the engineer responsible for the production of documentation is further removed from the effort late in the cycle. Whatever the cause of the documentation problem, there exists a need for a means of ensuring that the delivered circuits match the delivered documentation.

A part of the solution to these problems is to reverse-engineer the circuit. This has been done in several cases, but it too is expensive and time-consuming. In one case the Air Force reverse-engineered a circuit for the Navy. The project was done by staring into a microscope and drawing out the observed circuit by hand. The cost was \$75,000 and the project required 9 months. The result was a savings of \$22 million [Aviation]. A better approach is to automate the effort. This is the approach taken in ARES.

An automated reverse engineering system has an additional application. When working on a VLSI circuit, designers normally follow a Computer Aided Design (CAD) cycle. An example of such a cycle is shown in Figure 55. The cycle is usually entered with the requirements-specification; this is done with logic equations, or with some other type of formal algebra which defines the desired operation of the circuit. The next step is to define and implement in VHDL, or some other formal language, a particular solution to the requirements. This is a one-to-many mapping, with there being no one well-defined solution to any requirement. The VHDL, in addition to specifying a particular solution, allows that solution to be simulated and validated to ensure that the solution meets the established requirements. If it does not, either the solution is corrected or the requirements can be altered. In addition, methods are being developed to formally verify that the particular solution satisfies the requirements [Dukes]. After the VHDL is written and tested the particular solution is translated to an implementation in a particular technology. The translation can be done either through a mechanical means, such as a silicon compiler, or by hand, as in the case of full custom design. In either case, the translation provides a mapping from a particular solution into a technology-dependent solution. This mapping is also

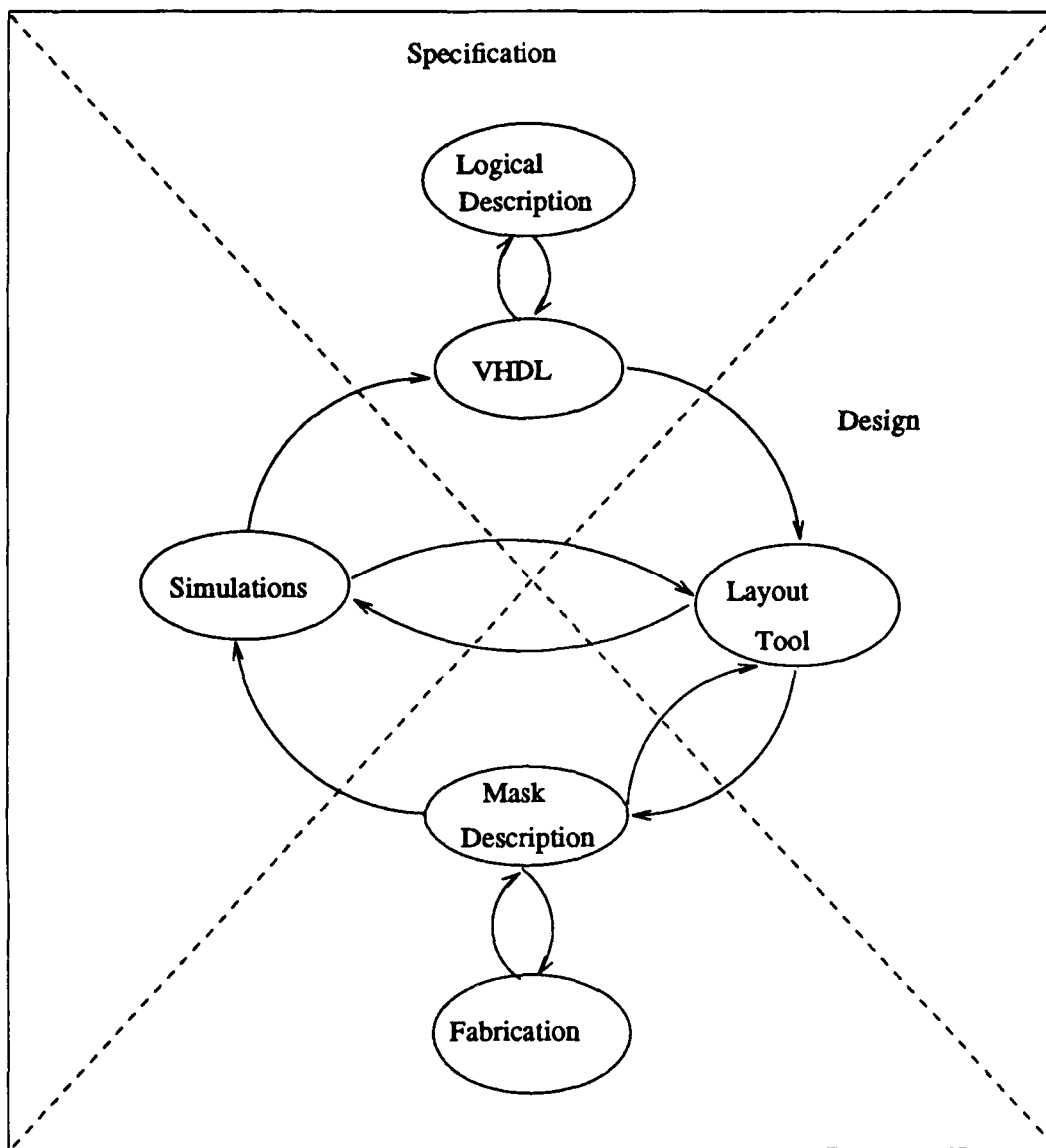


Figure 55: Typical CAD cycle

one-to-many, even within a particular technology, as the translation may be optimized for speed, area, a particular cell set, or any of a number of other cogent or incomprehensible reasons.

Once a design has been specified in a particular technology, the design is then translated into instructions for the actual production of the circuit. These may take the form of mask descriptions, or travel-directions to a laser injector. The circuit is fabricated according to these instructions. At either the level of fabrication-instructions, or the specification for a particular

technology, a netlist can be extracted from the design for simulation. This netlist is a description of each electrical component (transistors, resistors, capacitors, etc.) contained in the circuit and its interconnections. Using the netlist, the implementation of the circuit can be simulated to validate the design. Another option is to extract from the netlist a new VHDL description of the circuit. This description can either be compared to the original VHDL description, or be validated against the requirements [Dukes].

The CAD cycle is completed as often as needed to generate a "correct" circuit. The breakdown in the cycle occurs when the circuit is sent out for fabrication. At this point the circuit leaves the domain of the designer and enters that of the production and test engineers. These may not be, and in fact often are not, the same engineers who produce the designs. If an error has been found in the circuit once it has been sent out for fabrication, it can be difficult to fix the source of the error. In addition to problems with error-location the designer will point to errors in fabrication while the fabrication and test engineer indicates design flaws. Bringing the test phase back into the design cycle, and using the same tools for testing the fabrication as were used for designing the circuit, circumvents these problems and enhances the ability to predict incipient errors which could result during fabrication runs of a circuit.

ARES was conceived as a non-destructive, optically-based image processing system for reverse-engineering VLSI circuits. The system was chosen to be non-destructive because of the low number of samples available of any particular circuit which might need to be reverse-engineered, and to allow the use of reverse engineering as a point of re-entry into the design cycle. The validation of this approach to reverse engineering was done in a Master's Thesis by Fretheim [Fretheim]. That effort established the feasibility of automating reverse engineering and developed many of the required algorithms. The thesis made an initial attempt at establishing an overall system design, but this effort was hampered by the lack of theoretical bases. Follow-on efforts worked to parameterize and enhance the optical systems [Leano; Augustus]; to

empower the reasoning capabilities of the system [Hayden; Shoop] and to develop new image processing techniques [Mueller; Querns]. Further efforts in these areas are also underway.

4.2. Hardware Base

The CHIP system is currently hosted on two SUN workstations. The primary workstation is a SUN IV, which has been augmented with two CSPI Quickcard vector processor boards and a DataCube video system (Figure 56). A number of the routines in CHIP have been custom tailored for the vector processors, but if such boards are not available they can either be run using the vector processor emulation library, or by optimizing them to an available processor. The DataCube system consists of three RS-170 capture and display boards, a three-frame frame store, a graphics overlay generator, a multi-mode analog/digital image capture board and a region of interest store. The system is set up to handle the capture of grey scale, RGB, or digital camera images, and the display of grey scale, RGB, or pseudocolor outputs with overlays. A number of routines have been included in CHIP to use these features. Most of them can or have been set up to run on other image capture/display boards, or in the case of displays, directly on the system console. This workstation also has RS-232 serial links to the MITAS controller for the microscope stage and to an IBM PC which handles the capture of data for the photometers.

The other workstation is a SUN III which also has two CSPI Quickcard vector processors. This system has an ITEX FG-100 capture/display and image processing board (Figure 57). The FG-100 system is capable of accepting grey scale video from any one of up to three cameras. By multiplexing between cameras, it can also capture RGB video for still images. The FG-100 has on-board memory to store up to four 512 X 512 images, any one of which can be displayed at one time. By using 12 bits for onboard processing of images, the FG-100 can be used for overlay graphics. With the exception of the MITAS controller, digital-camera image capture, and the IBM PC connections, this workstation has the full capabilities of the SUN IV although at somewhat

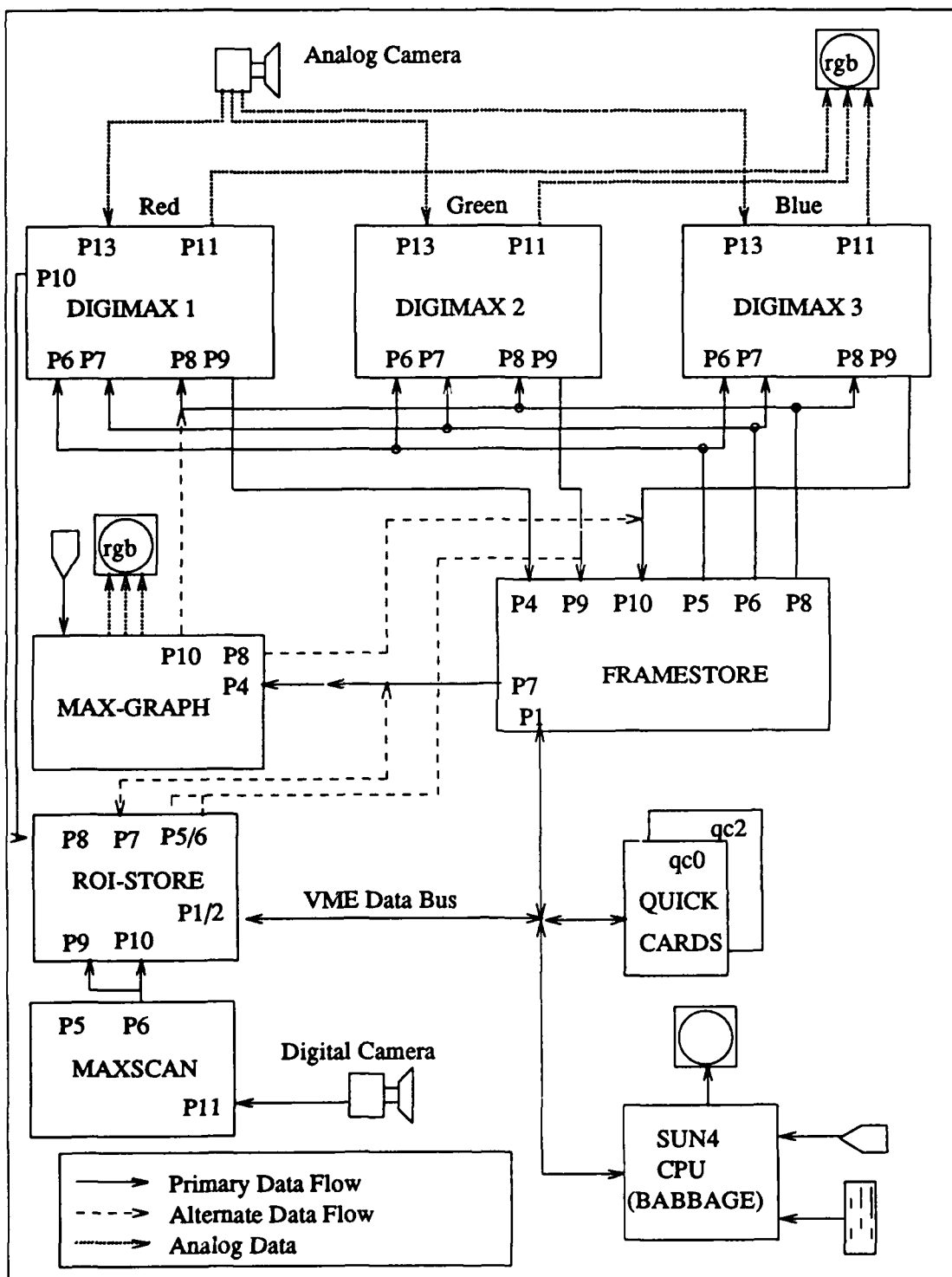


Figure 56: SUN IV System Configuration

reduced speed. There are RS- 232 connections available on this workstation, but perhaps a better option would be to access these devices over the network.

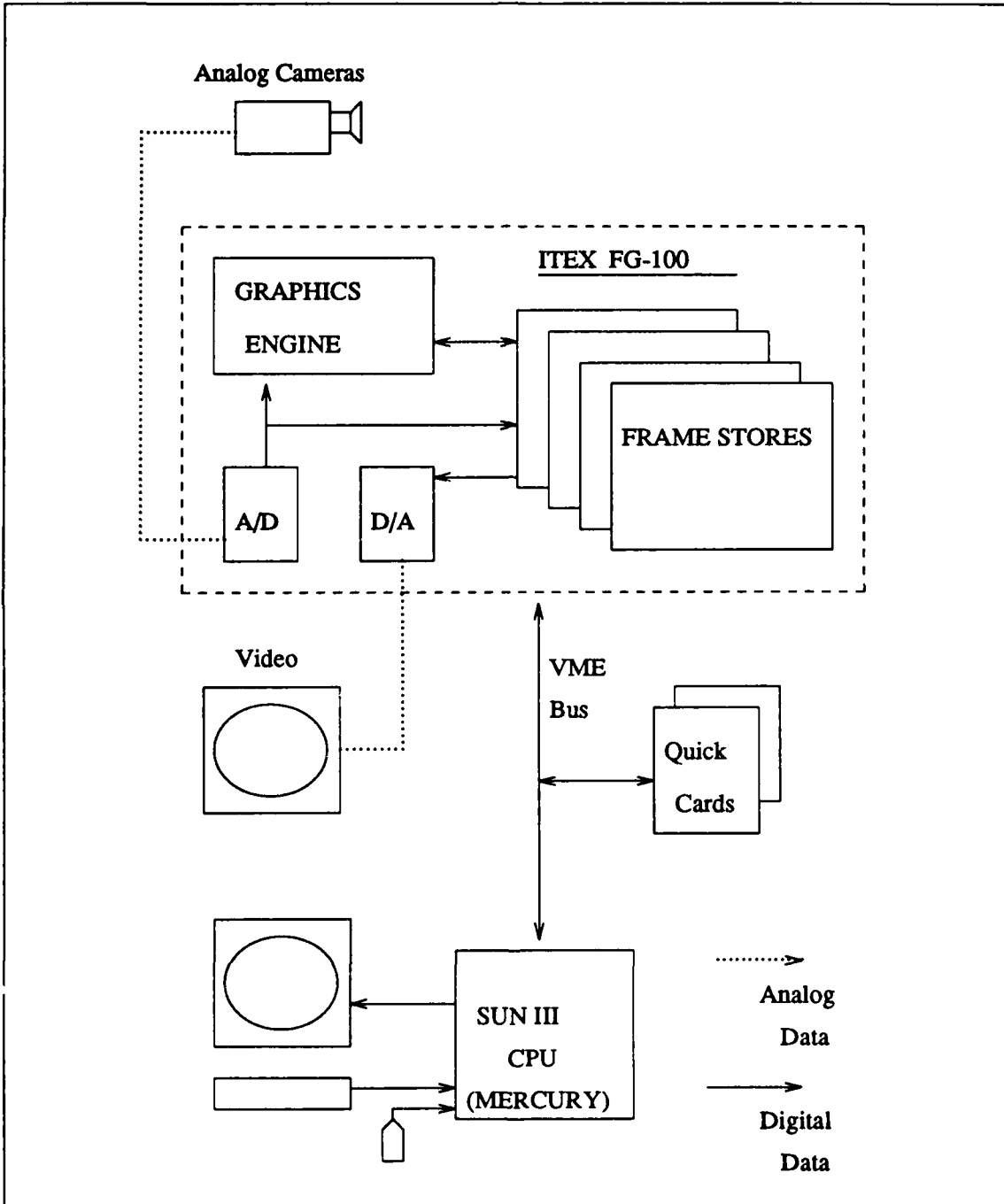


Figure 57: SUN III System Configuration

Images of VLSI circuits are produced using a videomicroscope. The microscope is especially designed for use with lasers and has two optical paths to the objective, one for use with lasers and one for use with incandescent sources. The laser path also has provisions for a white light spotter. The microscope has removable optics which can be changed to accommodate the particular frequencies of the laser being used at the moment. A viewing portal, mounted above the eye pieces, allows the use of either a camera or a photometer. The user is protected from receiving dangerous radiation in the eyes by an automatic cutoff which prevents discharge of the attached lasers whenever the eye ports are opened. The stage of the microscope can be driven in the X- and Y- directions by motors controlled from a MITAS controller. This setup is capable of submicron movements with a submicron repeatability over large portions of the movement window.

A wide variety of cameras is available, to include two black and white RS-170 cameras. One of these has low resolution and is used for guiding lasers and other such tasks. The other produces high resolution images and is used for actual image capture. An RGB camera is on hand to allow the capture of color images. The outputs of the analog cameras are digitized as 512 X 480 images. A digital high resolution black and white camera allows the capture of 1024 X 1024 images. This provides a higher resolution than is otherwise available at the same magnification.

Either a *HeNe* or a *CO₂* laser can be used with the microscope. The microscope optics and the photo detectors are changed to the requirements of the laser system in use. In addition, a variety of ancillary equipment is available to support experimentation.

4.3. CHIP

The platform on which ARES is built is a system called CHIP (CAD, Heuristics and Image Processing). This system is a set of tools which have been combined to make research in an

integrated environment possible. CHIP combines two externally developed systems, a CAD package and an expert system shell, with internally written image processing software. All of the portions are written in 'C', which has eased system integration.

The CAD portion of CHIP is the VLSI design tool, MAGIC, from the University of California, Berkeley. MAGIC is a graphically oriented interactive circuit editor. It incorporates a number of useful features including, but not limited to: interactive design rule checking, automatic routing, efficient database manipulations, multiple graphic menus, graphics support for SUN workstations and X windows, etc [Ousterhout]. Many of these features are important for the CHIP system; in addition the design decisions made to accommodate these features have proven important.

MAGIC serves as the world view for ARES. The MAGIC database format is capable of incorporating all relevant information about a circuit undergoing the reverse engineering process, within the limitation of a Manhattan architecture. This restriction is a result of MAGIC's internal representation structures. The database stores areas as maximally wide horizontal rectangles. Each of these rectangles is completely covered in one type of "paint" or layer. The layers are arranged on planes of mutually supportive layer types. The layers can represent either real or notional layers on a VLSI circuit, such as polysilicon - a real layer, or transistors - a notional layer, or they can represent some abstract data, for example errors. The rectangles are referred to as tiles; each tile contains information about the type, or types, of materials or data it represents, information about the tile-size, and pointers to the tiles it bounds at the upper right and lower left corners. The pointers are called stitches. They provide an efficient means of searching the area around a tile. The tiles on any one plane cover the entire plane without overlapping. The edges of the plane are bounded by special tiles which stretch into infinity.

Tiles are grouped on planes by types which interact. Typical planes include: metal-1, metal-2, active, and errors. Nearly every structural element in a VLSI circuit is completely contained within one plane. The contact cuts, which are electrical connections between layers,

always span the metal-1 plane and at least one other, as all contacts are made between first layer metal and the other layers. Thus a contact from second metal to polysilicon would first make contact from second metal to first metal and then from first metal to polysilicon. Keeping the types isolated on planes of interacting materials reduces the amount of checking which needs to be done among types and reduces the degree to which the layers need to be segmented, thereby reducing both processing and storage overhead.

The basic data structure in MAGIC is the cell. Each cell contains some header information and one of each type of plane (Figure 58). The cell also contains a special plane - the subcell

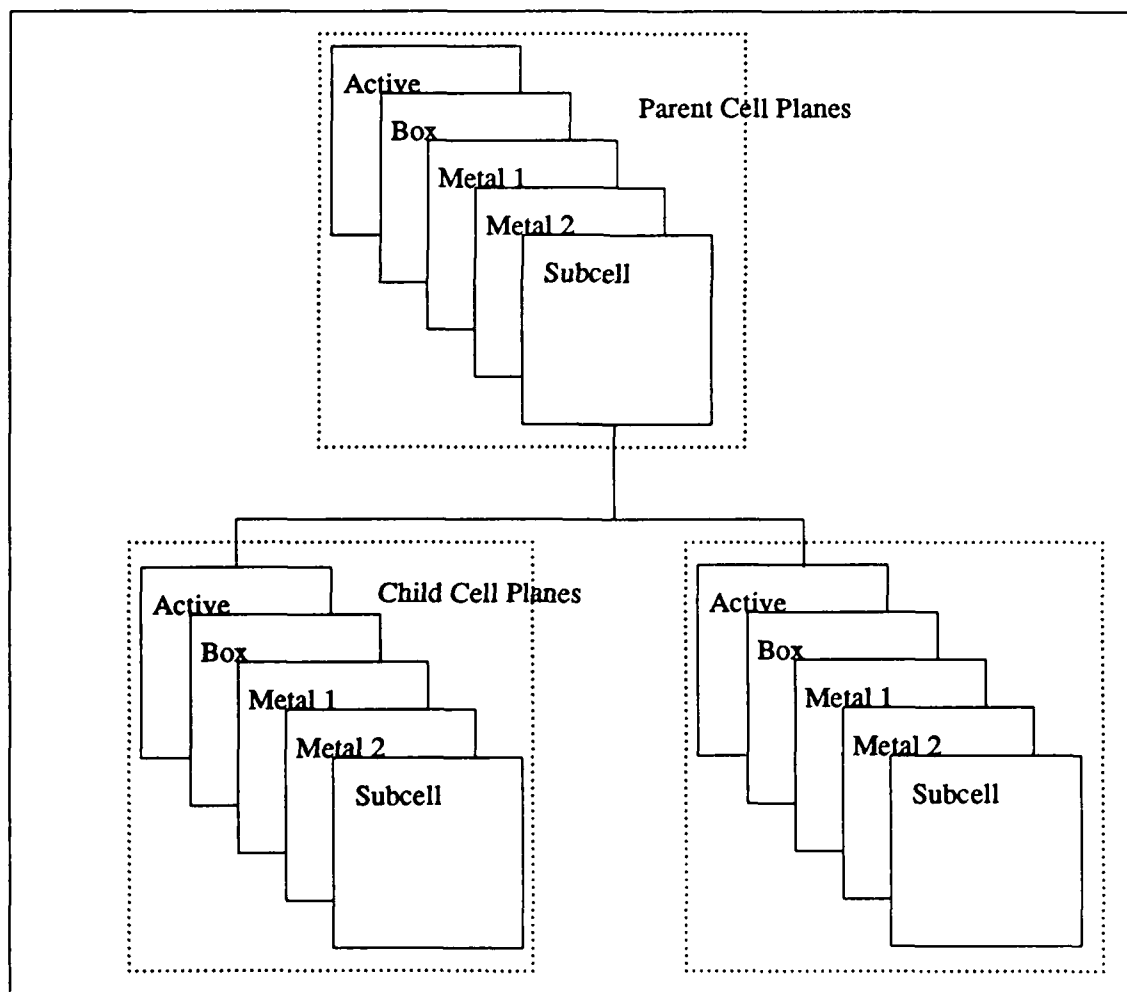


Figure 58: Cell Structure in MAGIC

plane. This plane contains tiles which map the size of, and contain pointers to, other cells included as subcells of the plane. The tiles in the planes of a cell do not interact with the tiles in a subcell of that cell. That is, they do not combine into maximally wide rectangles. This type of combination occurs only within each plane. Thus any subcell brings its unique structure complete into the parent cell. This proves useful in ARES as the subcells are used to represent areas of contiguous common composition. These areas are irregularly shaped regions where the composition of the circuit is consistent throughout the area. Each area is either all metal-1, all polysilicon, or all some other single layer or combination of layers. These areas are important both because determining the full extent of a material type is not always possible, and because the shapes of these areas give important information about the structure of the circuit. Having these layers represented as subcells allows them to be maintained as integral structures instead of having their form absorbed into maximally wide horizontal rectangles.

Information which is not distributed over an area can be stored by MAGIC as a label. Labels are placed in location and may be associated with a particular tile type. Labels can be used to mark such things as node names, important structures, etc., or to contain general information.

Information contained in MAGIC can be stored either as a MAGIC file, or in any of a number of common circuit representations. Significantly, sufficient information is contained in the MAGIC files that they can be translated into a netlist, or some other type of representation which can be used to either simulate, validate or verify the behavior of the circuit.

The world picture of ARES is an area of the MAGIC database, a number of image-like representations, state information and other data which describe the portion of the circuit which is currently under construction. The CHIP system provides a method to observe the progress of ARES, to provide user assistance to ARES, and to aid in the development of ARES. This system takes advantage of MAGIC's windowing capabilities.

MAGIC uses its own simple internal graphics and windowing systems. These systems are then translated into the graphics of the system on which it is hosted. This both eases the creation of windows and graphics, and allows portability over a wide variety of platforms. MAGIC has three basic window types: layout windows, netlist windows, and colormap windows. CHIP has added three windows to this system: CLIPS windows, image processing windows, and MITAS windows.

The layout windows provide a graphical representation for the information contained in MAGIC's database. The layers are represented as combinations of colors and patterns. The windows are also capable of accepting mouse or text commands from the operator.

The CLIPS windows provide an interface to the expert systems shell - CLIPS. CLIPS (C Language Integrated Production System) was developed by the NASA Johnson Space Center for use as an embedded expert systems shell [Giarratano]. CLIPS provides basic shell functions and a number of hooks for embedding CLIPS into a larger system. CLIPS rules consist of any number of clauses on the left and right hand sides of an arrow (Figure 59). Clauses on the left hand side are used to determine if the rule should be activated. If the rule is activated, it is placed on a stack to have the clauses on the right hand side executed. As each rule is executed others are checked to see if they should be added to or deleted from the stack. CLIPS has the capability to use external functions and data in its rules on both the left and right hand sides. This means CLIPS can access MAGIC's routines, or database, or the image processing routines.

CLIPS has a number of built-in functions to ease the development of expert systems, as well as system control functions. All but a few of these have been integrated into the CLIPS window (Figure 60). Those which were left out provided only small gains in capabilities in return for a major commitment of resources. The most frequently used CLIPS functions - run, clear, reset, watch, etc. - are implemented in a mouse-driven graphical environment. The watch commands are particularly important as they provide a means of observing the current state and flow of rule

```

(defrule store-lambert
  ?lam <- (lamberted ?x ?y)
  (need ?w ?u $?)
  (test (or (neq ?x ?w)
            (neq ?y ?u) ))
  (progressive-screen ?x ?y)
  ?process <- (processing lambert)
  (not (processing contacts/blocks))
  (or (not (blocks ?x ?y))
      (not (contacts ?x ?y)) )
  ?screen <- (current-screen ?l ?m)
=>
  (retract ?lam)
  (retract ?process)
  (retract ?screen)
  (assert (file lambert ?x ?y))
  (assert (need lambert ?x ?y))
  (assert (current-screen ?w ?u))
  (assert (need screen ?w ?u))
  (mitas-screen ?w ?u)
  (chipmenu "D chip.lam") )

```

Figure 59: Typical CLIPS rule

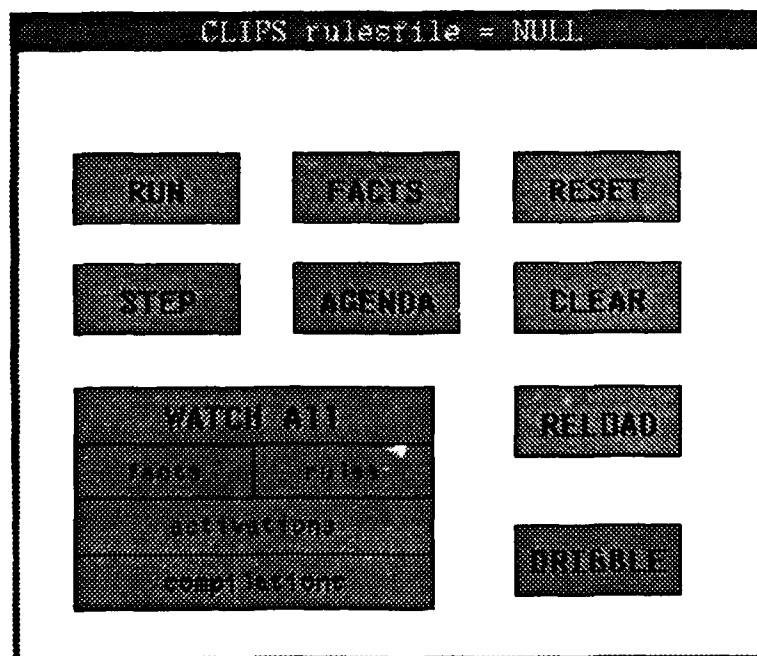


Figure 60: CLIPS Window

activations within the expert system. These and the other commands are also available from the MAGIC command line.

The MITAS window is used to control an interface with the MITAS controller for the microscope. This window has a mouse-driven menu to control stage movements (Figure 61). The MITAS controller commands can also be accessed from the MAGIC command line.

The image processing (chip) menu provides a means for accessing the CHIP system's image-processing routines. The image processing portion of CHIP is a set of menu-driven image processing routines. The chip window provides a few rudimentary functions, to include image display and capture, a pixel value histogram, and histogram equalization (Figure 62). All other operations are accessed either through the menus, or through function calls. Associated with the chip window are the video displays. These provide a means for viewing the results of image processing routines which are stored as "pix_rects", an image oriented format.

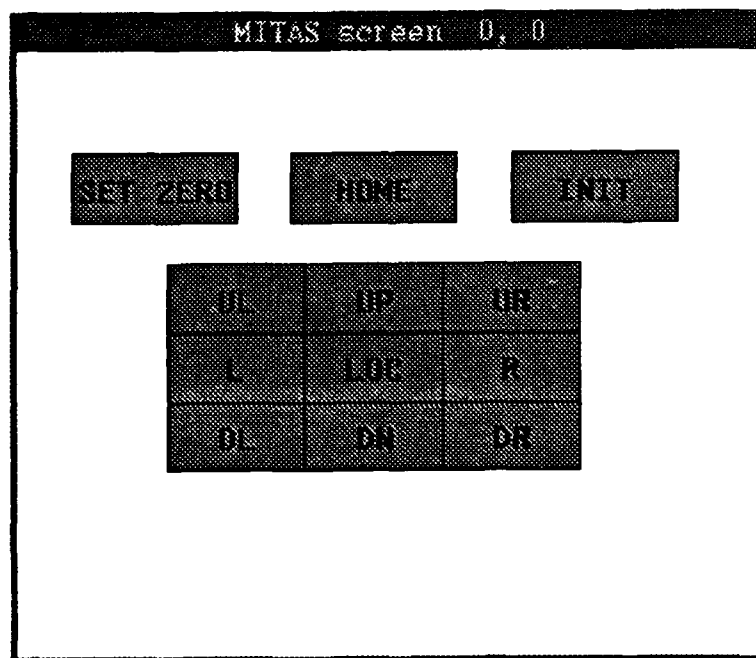


Figure 61: MITAS Window

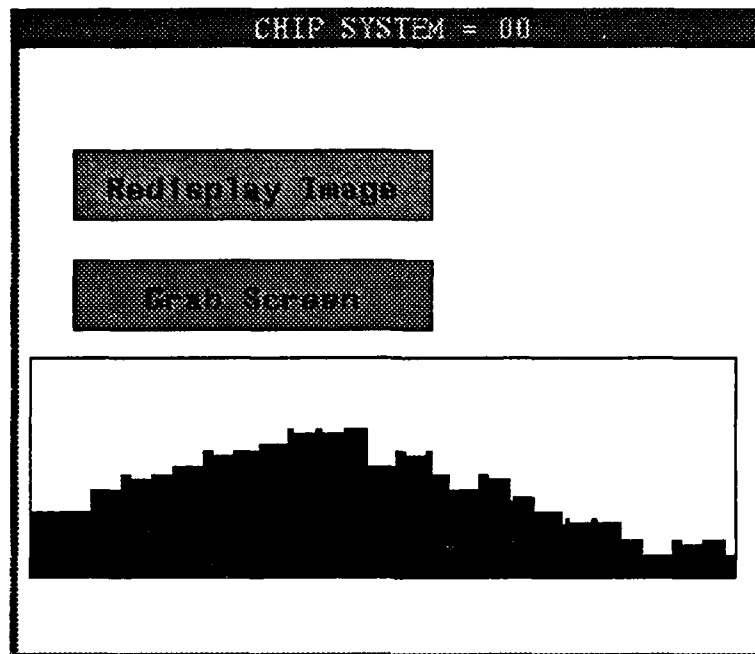


Figure 62: The Chip Menu

A User's Guide is available to explain the detailed operation of CHIP (Appendix C). In addition, there is a Programmer's Manual (Appendix D) which explains how to make additions to the CHIP system, and which sets standards for system development. These manuals are to be used in conjunction with the manuals for the other portions of the system (CLIPS Users Guide, CLIPS Reference Manual, CLIPS Architecture Manual, MAGIC Manual, MAGIC Tutorial, and MAGIC Maintainer's Manual).

4.4. Control

MAGIC's interactive design-rule checking has a number of useful design features. In order to allow users to continue working on their designs while MAGIC is performing design-rule checks, the design-rule checker steals cycles from the process between user inputs. As most designers do not work at speeds which press the limits of modern computers, any CAD system

must spend a great deal of time polling for inputs. By lengthening the time spent between polls and using the intervals to pick at a task, the process can accomplish an enormous amount of work during what would otherwise be wasted time. The process performs a portion of its task and then polls to see if there has been an input from the user. Because the process does not attempt to perform the entire design-rule checking at once, it appears to the user that there has been little or no delay in the system response. The system also can make use of this time for other purposes. This portion of the system is implemented as a loop in which the system checks a job queue to see if there are any tasks which need to be performed, and then polls for inputs. The client tasks are expected to internally limit the length of their processing, and return control to the main process. Inputs, when found, are also distributed to client processes. These processes then respond to the inputs and return control to the main process. By tapping into this system the expert system shell, and the image processing portions of CHIP, are able to obtain access to processing time and user instructions.

An expert system controls the internal flow of ARES by tracking which areas of the circuit have not been investigated and by checking for areas of the design which have already received an initial investigation, but about which the circuit builder has not been able to make all final determinations. If the control system determines that the area might be helped by additional information about the area, the controller requests that updated, more detailed or more concise data be gathered about an area. Otherwise, the controller might request that the system operator attend to the area. This can be done either by the operator's submitting more rules/facts to the circuit builder or by entering information about the area through the layout window. The controller requests more information from the user by painting the area in yellow dots.

The controller also determines which parts of the world picture and world view need to be placed into long-term storage. For example, a particular scene may be placed into storage either to allow another process to use that same scene at a later time, or to allow the processing on that

scene to be momentarily interrupted while the control system attends to some other area. The latter may be done if more information is needed about an area which had been processed earlier. In this case, the re-look would have priority over the initial processing of a new scene.

4.5. The Sensors and Input Processes

ARES uses several different sensor types. Each one is capable of providing some type of useful information about the circuit. There are a number of sensors which take advantage of the thin-film characteristics of VLSI circuits to enable the internal processes to determine the material composition of each area on the surface of the circuit. As yet there has not proven to be any one "ideal" sensor. As a result, the best approach is to interpret the results of each sensor in the areas in which it proves most useful, and to combine the findings. This is done by having each of the sensors independently write its results into the MAGIC database. An expert system "circuit builder" then combines the data into one unified result.

4.5.1. Lasers, Filters and Color

Due to the nature of their construction VLSI circuits exhibit certain thin-film characteristics. The construction of a VLSI circuit can take many forms. Circuit construction can be done by photolithography, or by some sort of ion beam, or by a laser-type injection process. At this time the photolithography techniques are somewhat more commercially viable and represent the majority of the circuits in use. I will now provide a general description of a typical process which demonstrates how these characteristics take shape. This description is not intended to cover all aspects of the process. The process uses a number of "masks" which describe where the materials will be placed.

The first step in a lithographic process is to clean and prepare the wafer on which the circuits will be constructed. After this has been done if the process is a single p-well process, a

layer of photoresist is spread across the surface of the wafer in all areas which will have p- wells. This is done using a negative mask of the p-well areas. The wafer is then bombarded with p-type impurities. Exposed areas of the circuit are now doped with p-type material. The areas which were covered in photoresist maintain an intrinsic composition. The photoresist is then cleared away and the circuit is readied for the next step. The next masks are positive masks of the active areas (n- and p-type diffusion) of the circuit. A layer of field oxide is grown across the surface of the circuit. This layer is stripped away in the areas where there are to be active layers. The active areas are then covered with a thin gate-oxide.

Once the gate-oxide has been grown, the wafer is covered with polysilicon. A mask is placed over the areas where polysilicon is desired and the remainder is stripped away. With the polysilicon in place the active areas are emplaced. By doing the processing in this order, self-aligning gates are formed. Even if the active dopants are slightly off registration they will still form a transistor gated by the polysilicon. If the active areas were laid first, it would take a lot more effort to get the polysilicon gate lined up exactly with the slot left for it. After the active materials have been put down another layer of oxide is added and then holes for contacts are cut through this layer. The first metal layer is then poured across the surface, and stripped away where it is not needed. The first metal layer is covered with another oxide layer and the appropriate cuts are made for metal-to-metal contacts. After the second metal layer has been poured and stripped away, a final protective coating of silicon dioxide, or overglass, is sputtered onto the circuit. Cuts for bonding-wires and probe pads are made in this layer and the wafer is ready for testing, slicing and mounting in packages [Geiger et al; DiGiacomo; Weste and Eshraghian].

The overglass can be approximated as a thin-film layer [Parthasarathy et al.]. Examining the equations for reflectance in thin films (eqns 6 - 9) we see that there are two factors in the equations which are affected by the processing of VLSI circuits. One of these is the depth of the thin-film layer, in this case the SiO_2 which covers the circuit. During the fabrication process a

number of SiO_2 layers are grown or deposited, but these fuse together and can be viewed as a single layer. This leaves for consideration only the interfaces between the SiO_2 and the underlying layers. As a result of the processing used to create the circuit, the intrinsic, n-type diffusion, and p-type diffusion layers have surfaces at approximately the same depths within the circuit. There will be some minor differences due to variations in the depths of the etchings, or due to the spread of the oxides deeper into the intrinsic material. The result is that we can expect these materials to have very similar appearances. These differences may, in some circuits, be small enough to fall within the noise of fluctuations in the depth of the overglass. In other circuits, they may be significant enough to allow segmentation of these layers. On the other hand, there are dramatic differences in the depths of the surfaces of the polysilicon and metal layers. As a result the differences should prove efficient for the segmentation of these layers.

Using thin-film equations from chapter 3, with $N_0 = 1$ (air), $N_1 = 1.5$ (SiO_2) and $N_2 = 3.4(1 - 9\sqrt{-1})$ (Si), we get the reflectivity curve shown in Figure 63. This curve predicts the ability to use white light to distinguish between layers based on the thickness of the overglass, which is a function of the layer depth. Here the white light would be considered to be spread over the response curve of the camera. This capability can be confirmed by observations of VLSI circuits. The gray scale image which results from a videomicrograph of a VLSI circuit illuminated with white, or nearly white, light has intensities which vary with the layer type. This can in itself provide a large amount of the information needed for segmenting the circuit into regions of distinct material composition. It is not, however, sufficient for all segmentation tasks. Not all circuits possess clear and distinctive intensity differences between material types. Further, segmentation based on white-light intensities alone is subject to distortions arising from unequal illumination of the circuit surface, variations in the image processing equipment, and noise from the roughness of the circuit's surface materials. That is, textures may make two otherwise distinct material types appear the same.

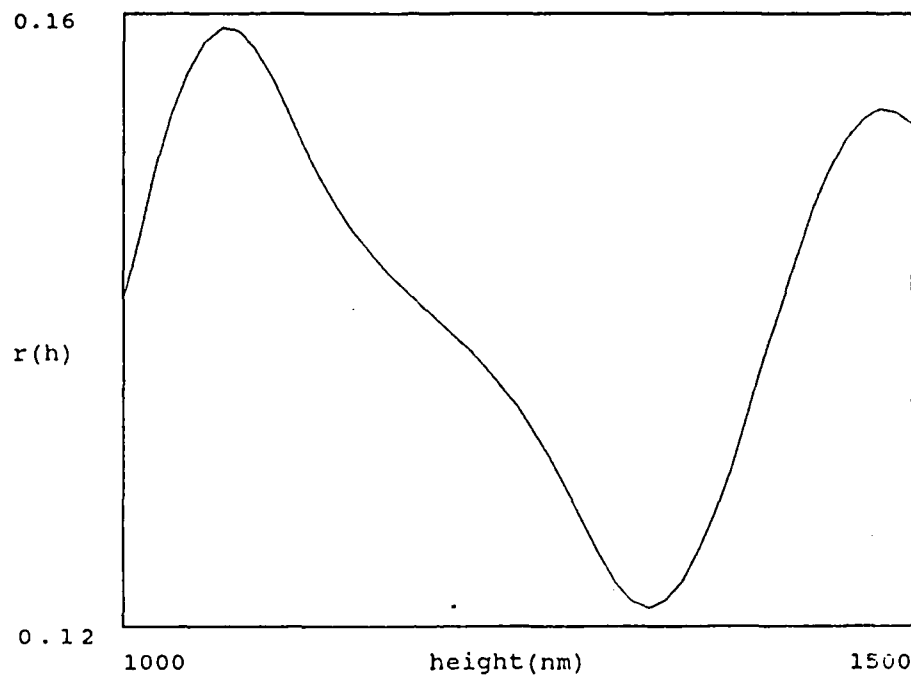


Figure 63: Expected Reflectivities for White Light

Using the color model of chapter 3, a trace of the film thickness plotted on a CIA chromaticity diagram appears in Figure 64. The accuracy of the model has been confirmed both by observation of the characteristic colors of VLSI circuits, and by comparison of the model to the observations of other researchers [Augustus]. The trace of the thicknesses can be adjusted by the appropriate filters as it is necessary to increase separation of particular film thicknesses.

The other thin-film factor affected during fabrication is the refraction coefficient of the underlying material. This varies with the concentration and characteristics of the dopants used in the process. Interestingly, the coefficients for intrinsic and both n- and p-doped silicon are nearly the same in the visible region. They do not begin to diverge until the near-IR portion of the spectrum, and reach the greatest separation at around 10 microns. This results in the reflection profiles shown in Figure 65. The 10 micron area is, notably, the region where the effective band gaps of the doped silicon layers are active. The band gap for intrinsic silicon is 1.1 eV, while that

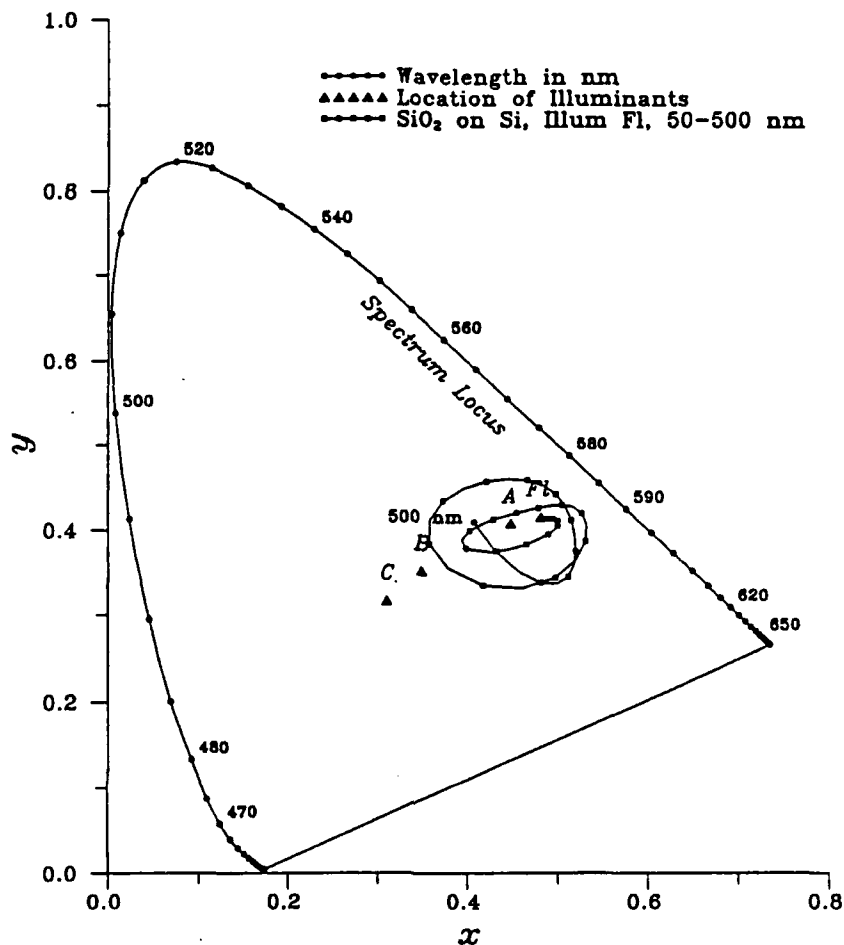


Figure 64: Chromaticity of Silicon Dioxide on Silicon for a Thickness Range of 50-500nm [Augustus, p. 44]

for p-type silicon is 0.045 eV. The photons of a CO_2 laser operating at 10.6 microns have an energy level of about 0.1 eV [Leano]. There will be a measurable difference in the absorption which is regulated by the concentration of the dopants. From this it is evident that, for circuits in general, to be able to reliably separate the intrinsic from the n- and p-type materials using a single frequency, or near-single frequency, it is necessary to operate in the IR region. There will be specific circuits for which these separations will be possible in the visible region.

Experimentation with *HeNe* lasers and filtered black and white cameras have confirmed these results. Augustus collected tables of reflectances which showed strong segmentation capa-

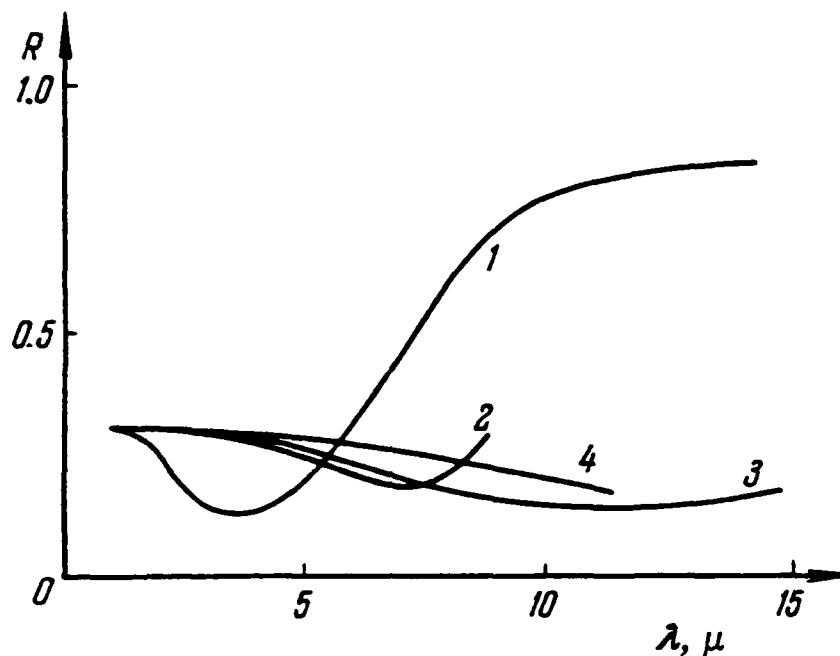


Figure 65: Reflection of Heavily Doped Silicon Samples: 1) $2 \cdot 10^{20} \text{ cm}^{-3}$ (B); 2) $5 \cdot 10^{19} \text{ cm}^{-3}$ (B); 3) $2 \cdot 10^{19} \text{ cm}^{-3}$ (As); 4) $1 \cdot 10^{18} \text{ cm}^{-3}$ (As) [Fistul, p. 237].

bilities for polysilicon and metal layers, but weak, though discernable, segmentation for n-, p- and intrinsic silicon layers (Table 4). The *HeNe* laser operates at 0.6328 microns. The light being reflected from the surface of a circuit and into a black and white camera was filtered with low-pass filters with cutoff frequencies of 0.830 and 1.0 microns. The result was an image in which the n- and p-doped areas of the image were clearly distinguishable. The problem with this approach was that the image quality was so poor that it was impossible to distinguish other features or the edges of the doped regions. Combined with other methods, however, this technique can be an effective means for distinguishing the locations of the doped areas.

4.5.2. Color Manipulations

A first response to working with the color images obtained is to attempt to separate the layers by grouping neighbors in the three-dimensional color space. The first step for this process

Table 4: Results of Laser Segmentation

Layer	Test Chip	# of Meas	Correct	% Correct	Modal Filter % Correct	Identified as
Metal1	1	100	98	98	98	Metal1
	1	100	73	73	83	Metal1
	2	100	81	81	87	Metal1
	2	25	24	96	100	Metal1
	2	100	66	66	83	Metal1
Metal2	1	100	100	100	100	Metal2
	2	25	18	72	88	Metal2
Intrinsic	1	100	54	54	60	Intrinsic
	1	100	10	10	0	P-Diffusion
	2	100	25	25	12	P-Diffusion
	2	25	5	20	8	P-Diffusion
Polysilicon	1	100	80	80	89	Polysilicon
	1	100	72	72	95	Polysilicon
	2	25	18	72	100	Polysilicon
N-Diffusion	1	100	94	94	98	N-Diffusion
	1	100	74	74	98	N-Diffusion
	2	100	99	99	100	N-Diffusion
	2	25	23	92	100	N-Diffusion
P-Diffusion	1	100	81	81	93	P-Diffusion
	1	100	62	62	64	P-Diffusion
	2	25	9	36	44	Polysilicon
	2	25	0	0	0	N-Diffusion
Nfet	1	100	77	77	84	Nfet
	2	25	12	48	68	Nfet
Pfet	1	100	26	26	15	Nfet
	2	25	12	48	72	Pfet

is to obtain samples for each of the designated layer types. The mean three-space location for points in the sample area for each material is formed as the set of means along each of the color axes. The standard-deviation estimator is calculated as the average vector distance from the sample mean color-value to the color value of each point in the sample population. After obtaining the sample mean, the image is then searched for all pixels with a color value within a designated vector distance of the mean color value of the sample. The best results have been achieved by using three times the standard deviation estimator as the designated vector distance. The algorithm for assigning layer values checks blocks $1/\lambda$ in size, where λ represents one half of the size of the smallest feature on the circuit. λ can be designated by the operator, or can be determined automatically through the use of Cepstrum analysis [Fretheim; Fretheim and

Kabrisky]. If the contents of a 1 lambda square contain more than 50% pixels within the designated distance of the nominal color value, the square is "painted" into the MAGIC database as a square of that material type (Figure 66). The data are now ready for use by the circuit builder.

Another option which can be used when the coloring is distinctive, but there are large illumination differences over the surface of the image, is to normalize the color values. This is done by dividing for each pixel, the red, green, and blue components of the color by the square root of the sum of the squares of the component values. The result is a new image in which pixel values are dependent on color only and not intensity. This method was not as effective as a non-normalized extraction technique for the tested images of circuits.

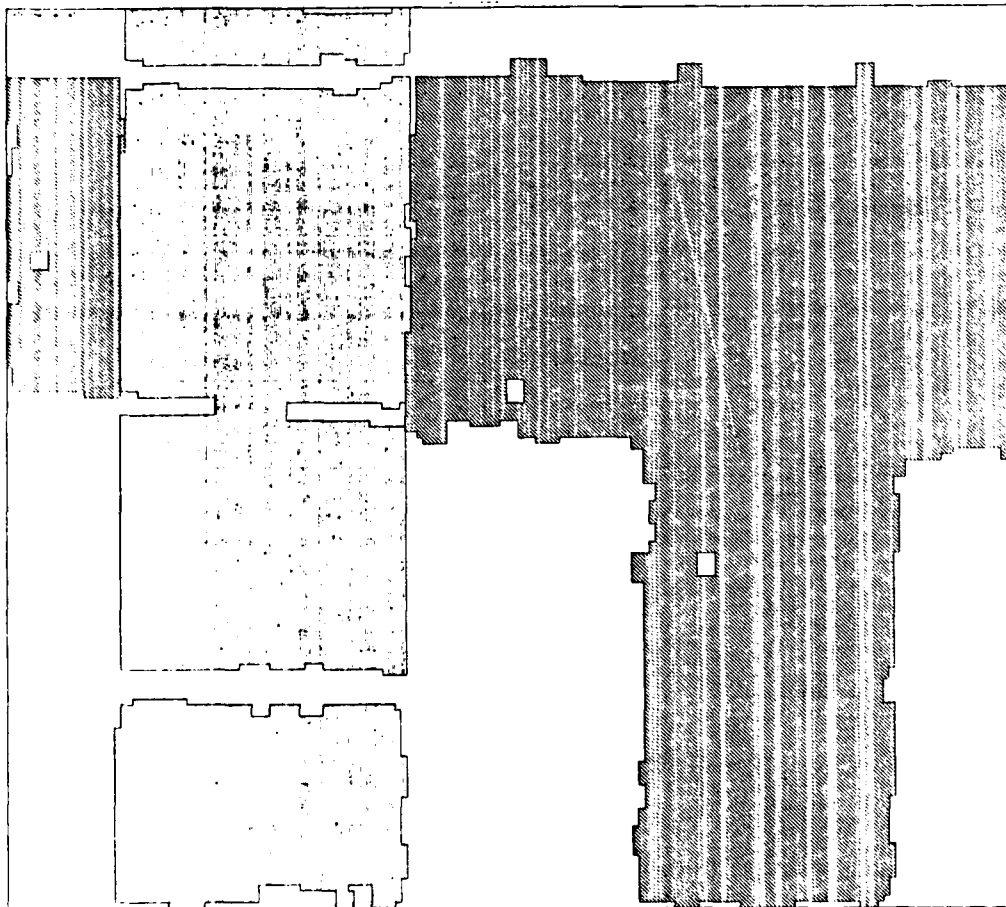


Figure 66: Results of Separation in 3 Color Space; Metal 2 and Polysilicon Shown

A third means of dealing with the RGB camera outputs is to convert them to an opponent color system. This is done using equation 18 and a variation on 19:

$$R^+G^- = R + G \quad (23)$$

$$B-Y^+ = B(\lambda) + 0.5[G(\lambda) + R(\lambda)] \quad (24)$$

where

$$R(\lambda) = \frac{Red}{\sqrt{(Blue^2 + Red^2 + Green^2)}}$$

$$G(\lambda) = \frac{Green}{\sqrt{(Blue^2 + Red^2 + Green^2)}}$$

$$B(\lambda) = \frac{Blue}{\sqrt{(Blue^2 + Red^2 + Green^2)}}$$

The resulting R^+G^- and $B-Y^+$ for the same image used to obtain Figure 54 are shown in Figure 67. Using the same algorithm that produced the results of Figure 66, where the third plane used is intensity, will produce similar results to those of Figure 66. This is however an extremely inefficient method. Similar results can also be obtained by using a modification of this algorithm on just the two opponent planes. Another possible method would be to pick one of the planes on which the particular material is best segregated and choose some algorithm to recover information from that plane. Perhaps one of the better choices would be the use of the Queen Victoria Algorithm.

The Queen Victoria algorithm is a non-causal, non-linear, heuristic filter. It was first developed in 1985 by Captain J. Holten [Holten] and was later improved upon by Captain R. Roberts [Roberts]. The algorithm has been used in other reverse-engineering efforts to separate layers from grey scale images [Fretheim]. The Queen Victoria Algorithm scans an image a line at a time. Each pixel is converted to a symbol which designates whether the pixel is a part of a

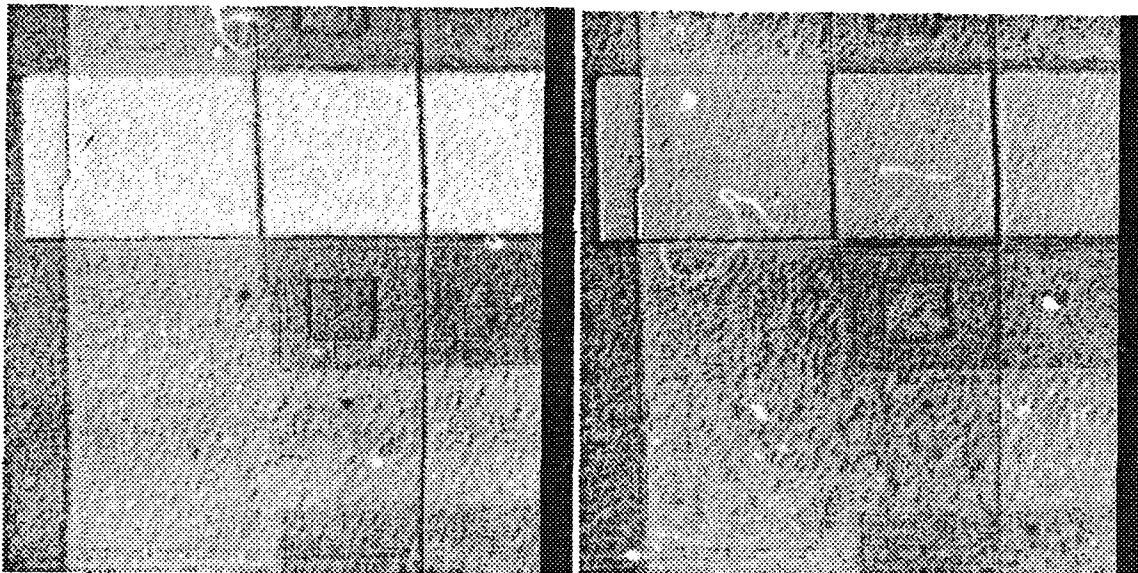


Figure 67: Red-Green and Blue-Yellow Opponent Planes

flat area, a potential edge, or a gradient region. A set of production rules is used to convert the gradient symbols to either edge or flat-area symbols. The line is then scanned a second time to replace the symbols with pixel values common to entire flat regions. This procedure is applied to both horizontal and then vertical lines from the image. It is repeated a number of times to allow convergence to a final image. Five applications generally allow for convergence. When the processing is complete, the image is divided into regions of constant value. The variations in the image due to noise are reduced or eliminated.

Applying the Queen Victoria Algorithm to the $R+G^-$ after low-pass filtering gives an image where the metal 2 region is represented with a single pixel value. From this point it is a simple matter to strip off the metal 2 region. Similarly, the $B-Y^+$ plane can be processed to obtain a single value representing both the polysilicon and metal 2 portions of the image. The metal 2 region

found in the other plane can be subtracted from this image and the resulting area extracted as polysilicon.

The color opponent planes can also be used in all other image processing applications either individually or in tandem with the other opponent, or the intensity plane.

4.5.3. Edge and Region Locations

One important step in being able to successfully reverse-engineer a VLSI circuit, or for any other vision system, is to be able to divide the chip surface into areas of common composition. This helps the overall task by simplifying it into the task of deciding the material composition of each block. The segmentation of the chip surface has been approached in two different manners, with a region-growing technique [Fretheim] and by a split and merge performed on quad trees [Querns]. Either method makes use of information about the interiors of regions and edges. Either can be improved with the provision of more information about edges.

One way to provide improved edge information is through the use of flow diagrams created by using Gabor filters. Because the features on VLSI circuits are primarily horizontally and vertically oriented, using the simpler form of subtracting the horizontal from the vertical component provides an adequate determination of the primary direction of any edge components, and at the same time provides information on the relative strengths of the edges (Figure 68a). This diagram can then be incorporated into the calculations of the region-determination algorithm, or it can be further enhanced to provide a higher-stage edge detection. A simple way of accomplishing this is to detect where the extreme values of the image lie. This results in a simple line drawing of the image (Figure 68b), which can now be further processed by morphological or other operations.

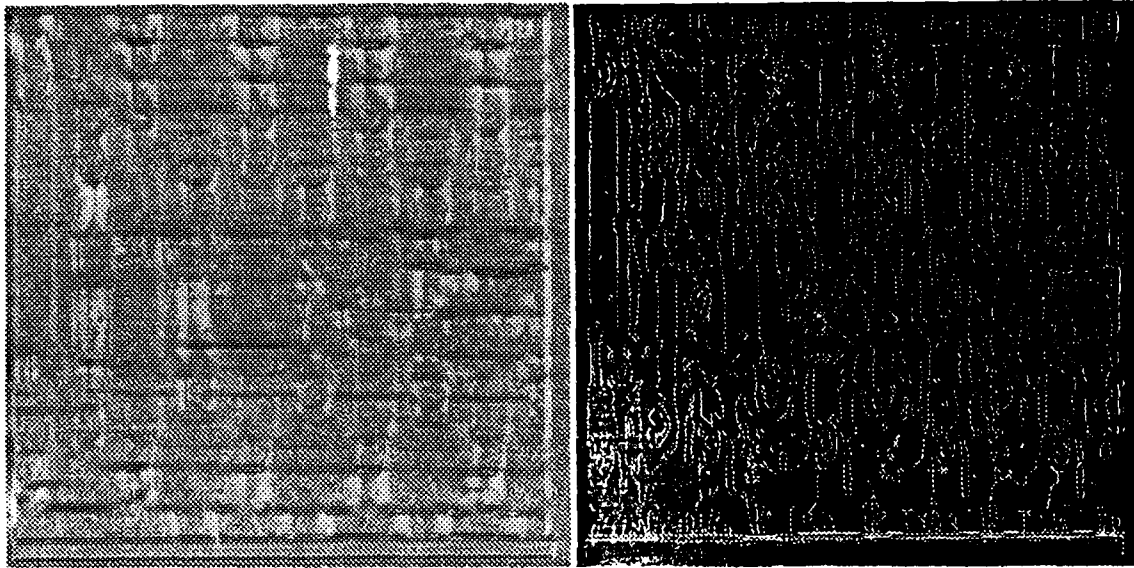


Figure 68: Edge Enhancement of an Image: A) Flow Diagram. B) Line Drawing

4.5.4. Contact Finding

One of the dominant features on the surface of a VLSI circuit is the contacts. These are the areas where the metal layer is allowed to come into contact with either the underlying semiconductor layers, or with an overlaying metal layer. The contacts are formed by cutting a hole in the silicon-dioxide which separates the two layers. The metal is spread on the upper surface. As it approaches the cut which was made for the contact, the thin layer of metal conformally coats the hole. This coating process causes the metal to fill in the corners of the contacts. The corners become rounded. In a process with very small feature sizes and square contact cuts, the resulting contacts appear to be donut shaped. In other technologies, the contacts may not appear entirely circular, but they do maintain their distinctive rounded corners. Transistors, metal lines, and other features are dependent not on their shape, but on other properties. The contacts, as a feature on VLSI circuits, are unique. They are the one feature which has a consistent shape and size. Con-

tacts are also very important in establishing the circuit description for they provide important clues as to how the parts of the circuit are joined together. Almost every transistor has some type of contact associated with it. In addition, there are distinctive arrangements which imply certain circuit features.

The distinctive signature of the contact makes it a good candidate for identification through correlative techniques, among which we include back-propagation networks. These techniques perform their function by looking at individual sections of a scene and correlating it with the object for which they are searching. This may be done a scene at a time through Fourier transforms and correlation in the frequency domain, or it may be done step by step by feeding each small area of the scene into the bottom of a neural network. Either way it consumes a great degree of computational power. The computations can be reduced by finding a technique to focus the attention of the system on only those areas where the contacts are most likely to be found. One technique is the use of Gabor filters as an attention mechanism. This technique has an additional advantage, in that the Gabor filters also form an excellent recognition feature set.

The attention processing is performed by using Gabor transforms on a scene from a chip surface. Because the contacts are circular, but the majority of features on a chip's surface are oriented horizontally and vertically, the Gabor filters used avoid these orientations. By being somewhat tilted, the Gabor filters that are used are able to deemphasize other features on the circuit, and yet they respond very well to the circular sides of the contacts. The actual Gabor orientations used are 20, 45, and 70 degrees, with a secondary set at 110, 135 and 160 degrees. The secondary set is not required for all images.

The size of the Gabor filters is chosen to correspond to the expected size of the contacts. Because the size of contacts does not vary across the surface of a single chip, this only needs to be set once per circuit. The Gabor filters are sized such that the edges of the contacts fall two standard deviations from the center of the wavelet. This matches the majority of the information

about a contact to the most active area of the filter. Experimentation has shown that the most effective filters are sine-wave Gabors with 2 cycles per envelope. The filters can also be applied with a decimation of 2, thereby reducing the overall computation by a factor of four. The results of applying a single orientation are shown in Figure 69. Some high responses are seen in the areas of the contacts, but nothing distinctly significant.

After the Gabor transforms have been applied to a circuit, the results from each of the individual transforms must be gathered to be used in some meaningful way. In the transformed images, the most important information is contained in the peaks and valleys (those areas where the image correlated most strongly, or most negatively with the Gabor filters). The method of combining the results from the transforms should preserve this information. One way to accomplish this is to take, for each pixel from the set of transformed images, the most highly responding

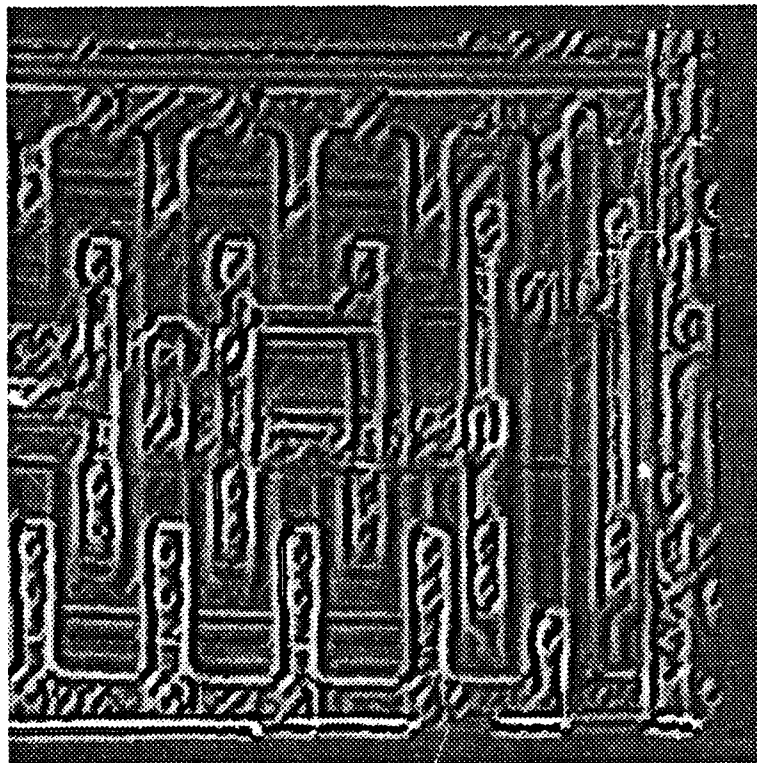


Figure 69: Gabor transform applied to circuit

value - negative or positive - and place that value into a new image. This new image then represents the conglomeration of the most important information from each of the transformed images (Figure 70). In this combined image, the contacts are distinctively highlighted. Note that this process, a non-linear thresholding operation, cannot be modeled by any usual formal filter description.

The contacts emerge as among the areas in this new scene with the highest pixel values. In order to reduce the time spent searching for them, the scene can be thresholded to indicate those areas in which contacts might be present (Figure 71). The areas remaining in the thresholded image are expanded slightly to insure that the centers of the contacts are included in a reduced search area. There will be a number of areas with high energy in the same spatial-frequency range as the contacts, but the overall area of the scene which will need to be searched will be

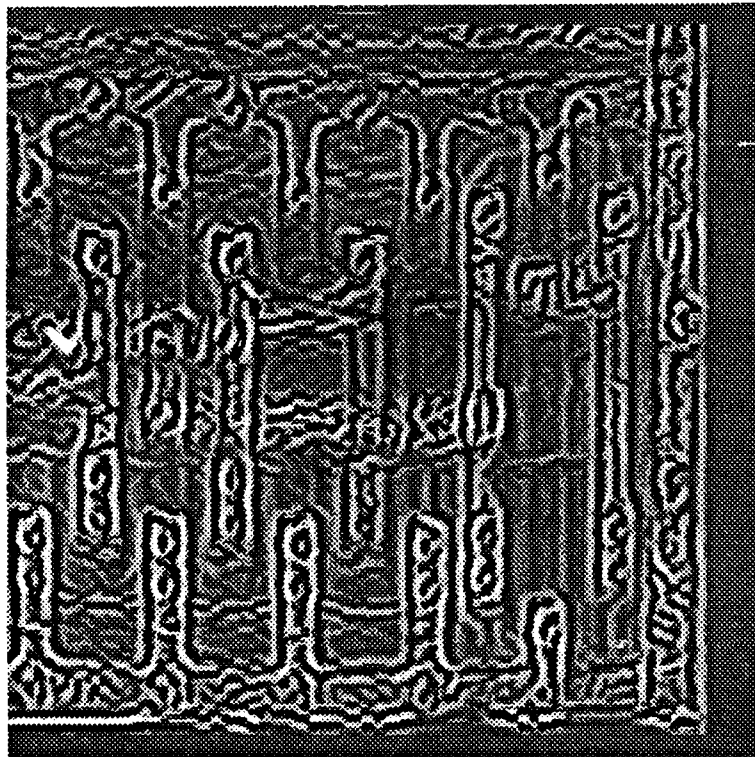


Figure 70: Combined Image

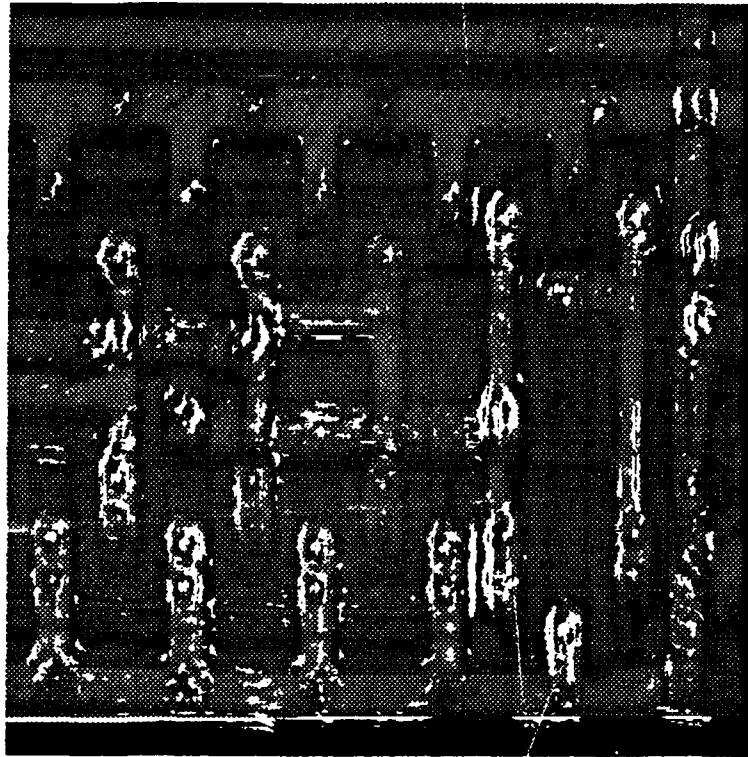


Figure 71: Image Thresholded to Reveal Likely Contact Locations

reduced by as much as 90% or more (Table 5). The search area will reduce itself even more as many of the high pixels in the search mask are grouped together, and once a contact is located in a particular area the remainder need not be searched.

Table 5: Reduction of Contact Search Space

Chip	Scene	Contacts Present	Contacts Detected	Area Covered
-				
A	1	27	27	8.0%
A	2	45	45	7.2%
B	1	42	42	6.4%
B	2	54	54	8.3%
C	1	23	23	3.6%
C	2	11	11	0.5%
C	3	24	24	14.6%

Once the areas in which to search have been identified, the back-propagation network is used to identify the actual locations of the contacts. The network has an input layer, an output layer and two hidden layers. The input layer has 256 nodes. Each node is fed the value of a pixel from the composite transformed image. Every second pixel is used from a 32 X 32 pixel area surrounding the center point. This is slightly larger than the size of a contact, but allows for the network to consider the outer edges of contacts in deciding whether one is present. The hidden layers have 64 and 8 nodes respectively. The top node has just two nodes, one which signals the presence of a contact and one the absence.

The network is trained using an initial scene, or scenes, from a circuit. A number of contacts in the scene(s) are identified as well as several representative regions which do not contain contacts. The training contacts are chosen to be representative of the variations in contacts found on the circuit. The non-contact locations should include any areas which potentially could appear similar to contacts. When training, the network will usually converge in around 5,000 cycles. This is rather rapid for a back-propagation network. In some tests the network has converged in less than 2,000 cycles. Other tests have required as many as 15,000 cycles. Results of testing with the back-propagation network are given in Table 6. In general, the contacts not found were those which people, even those trained in studying images of VLSI circuits, also had trouble distinguishing¹. The same network, when fed raw video to its input nodes, is unable to converge. This indicates the quality of the Gabor-filtered composite image as a feature vector.

Table 6: Results of Backpropagation Classification

Chip	Scene	Contacts Present	Contacts Found	False Alarms
A	2	45	44	2
C	1	23	23	1

¹One interesting result was that in one scene the back propagation network identified more contacts than did the operator. This was because the operator had not counted a row of contacts which were only partially included in the image.

167

4.6. Internal Transforms

Once the materials, features and areas of continuous common composition have been located and entered into the MAGIC database some work must be done to assemble the data into an actual circuit. This is done both through internal features of MAGIC, and through logical inferences made by an expert system - the circuit builder. The circuit builder takes the layer data and expands each layer to fill all blocks of continuous common composition which contain a portion of that material. There may be some blocks with only a slight indication that they contain any particular material, but under the current system they are completely filled. In a future enhancement it would be desirable if a slight presence of a particular material in a block were an indicator to either look for more of that material, or to recheck the boundaries of the box. Having filled all of the blocks with what it can, the circuit builder then begins to search for materials which are not physically represented, but which can be inferred from the presence of other materials and structures. As it proceeds it indicates areas in which it has difficulty. This information is passed back to the system controller which decides which further actions need be taken.

Much of the input for the circuit builder comes from MAGIC routines and features. This is accomplished by using the C-language interface for CLIPS. The MAGIC design-rule checker reads a file of "rules" for the design of circuits in a particular technology. The circuit builder of ARES uses an adapted subset of these rules, as well as other more general rules, to determine the makeup of a particular circuit. A better approach is to use these "rules" directly, as well as to make use of the results of the error checker. The error checker also has a useful paradigm of "growing" bounding boxes to determine the extent of the area which it needs to check for errors.

As additional benefits, search, database manipulation, and many geometrical routines are available within MAGIC, so there has been no need to write new routines to accomplish these tasks. MAGIC even has the capabilities to perform some of the simpler tasks, such as the combination of several layers into another due to new information. For example, if polysilicon is found

in one area and later diffusion is identified in that same area, when the diffusion is written to the database over the top of the polysilicon, the materials will be combined to form a transistor.

CHAPTER 5

Conclusions and Recommendations

5.1. Conclusions

The work covered in the last four chapters has been a somewhat bewildering journey from metaphysics to the physics of thin glass films with diversions into neural networks, wavelets and other areas along the way. The focus throughout this journey has been on the functional elements of a machine vision system and their intricate relationships. Two facts stand clear. First, vision requires an extremely complex system. Second, the individual components of the system can be relatively simple. This underscores the importance of the interactions among the system components.

5.1.1. The Vision System Model

The vision system model represents an important step in the development of pattern recognition and vision systems. It defines the requirements for a vision system and provides a foundation upon which the system can be built. By having a model, the task of building a vision system is simplified to one of finding the proper implementations of the particular parts. Not using an established model means that development of the vision system proceeds with no clear concept of the direction in which to proceed, or of the scope of what is required to build the system. That is not to say that every system will require a full implementation of the model, but in every case, the model can provide the basic system structure.

Included in the vision system model are a number of important concepts. One of these is the emphasis of the model on the need for feedback within the system. Another is the centrally

focused control which results from building a world picture within a world view. Focusing the processing on the goal of building the world view and picture coordinates the efforts of the system in an efficient manner. It serves to avoid distractions, ease the integration of multiple sensors and reduce the need for context switching. In this manner it also serves to reduce requirements for memory, processing and other scarce resources. This goal also allows the control section to reduce the amount of direction it needs to provide, thereby reducing control requirements and allowing greater flexibility to the constituent portions of the system.

The use of multiple pathways, both in processing data from the sensors, and in processing and updating data in the world picture, is important in that it allows a synergistic effect whereby results emerge from the interaction of multiple processes, rather than being the culmination of some set stack of routines. This also allows multiple uses of intermediate results, a gain in efficiency.

Another important portion of the vision system is the concept of a super-conscious. This gives a somewhat more "human" quality to the vision system. Rather than relying entirely on analytical methods, the vision system is able to proceed with intuitive leaps. This is a major gain.

5.1.2. Gabor Filters

The capabilities of the Gabor filter to perform processes found in biological models goes beyond the responses of individual neurons in the visual cortex. Using Gabor filters allows the duplication of a large number of optical illusions. The existence of these illusions exposes the raw edges of the processing system that nature uses. In duplicating the results obtained at these raw edges and by at the same time being able to imitate the gross processing abilities of the system, we can confirm the validity of Gabor filters as a model for biological vision processing.

A most important concept is that of Gabor filters as a model for the base-attentional mechanism. This has implications both for further research into attentional processes, and in the

design and understanding of systems which either require, or wish to avoid, attention-arousing mechanisms. The high correlation of fixation points indicates that the search process involves a search strategy sorting among attentional indicators. This is a much firmer understanding than saying that people look for "interesting" features. By using Gabor filters the "interest" of a feature can be directly measured. Once this measurement is taken, it allows a deeper understanding of the true search strategy, and why a particular feature was chosen over another seemingly similar feature. Knowing the basic attentional indicator also allows a measure to be put on how influential a particular feature might be in attracting attention. The attentional indicators, of course, must be considered in conjunction with particular search strategies. The measurements can be put to use to either improve the visibility of essential controls, indicators, lights, and other objects, or to aid in detection avoidance. The results are improved safety, quicker responses, and better camouflage.

Another area in which Gabor filters show considerable promise is in the detection of edges. This is especially evident when the Gabor correlation results are used to create a flow diagram. These diagrams display the essence of edges - slope orientations. For the creation of these diagrams, Gabor filters are far superior to other techniques, as they consider not only the immediate pixels, but also the surrounding neighborhood. This gives a more accurate description of an edge, and it allows discrimination in which edges the system will respond to. Further, edges can be determined on a localized basis and don't require global-line-support-region calculations.

Gabor filters have also proven to provide a very good feature-set for object recognition. The fact that Gabor filters can be used to represent the response of cells in the visual cortex suggests that they should have utility as feature vectors in recognition problems much more complex than the rather simple demonstration presented here. In fact, the flexibility of the Gabor filters and their ability to provide an optimal spatial - frequency representation suggests that they should provide, if not the optimal, the most general feature representation possible.

5.1.3. The Color Model

There are three important concepts involved in the color model. The first of these is to be able to predict the color and separability of regions of an image. The second is to enhance the separation of areas through the use of filters to alter the color space in which they appear. The third important concept is to map the color representation into an opponent color system. The combination of these enhances the ability to use color for image processing.

The ability to predict color, even under limiting conditions, is significant in that it allows for the measurement of material properties. It also provides the basis for a method by which colors can be altered and thus allows for segmentation by selected properties. Although segmentation can be done without prior knowledge of the particular colors which will be expected, any procedures for improving the segmentation through altering the perceived colors must be done on an ad hoc basis. However, with an accurate predictive model, the conditions to alter the perceived colorings can also be determined.

Converting the pure color representations into an opponent color scheme allows for rapid determination of color changes without regard to color intensity. Any differences in the opponent color planes directly represent a difference in the color of the object. There is no need to calculate normalized vector differences in a three-space. The opponent planes also provide an immediate representation of color edges. The edges can be determined using the same methods as can edges in a grey-scale image. Comparisons and inclusions of information for color data are simplified to direct manipulations from one grey-scale image to another with no conversions from three-space.

5.2. Recommendations

The definition of a vision system model opens the way for a wide variety of research possibilities. In addition to modifications to and improvements upon the model itself, there is work to

be done on each of the system components. There are also a number of related areas which have arisen in the course of developing the vision system model.

5.2.1. Attention and Search Strategies

The use of Gabor filters as an attentional mechanism provides a valuable insight into the functioning of the human visual system, even if it is limited to grey-scale, still images. Even more valuable would be an explanation which also incorporates motion and color. Gabor filters as attentional indicators could be extended to cover these areas. Several researchers have demonstrated the potential for using Gabor filters, or a wavelet scheme, to process information about motion [Adelson and Bergen; Emerson et al.]. It is a reasonably direct step to construct an attentional mechanism for the work they have accomplished. Further, a simple attentional system can be created for color by using Gabor filters and an opponent color system. Once these models have been established their predictions need to be checked for accuracy by comparison to actual human responses. Two problems arise in this respect, both of which are more difficult with respect to motion. The first problem is the increased processing required to provide predictions. Requirements for color processing are only three times that of the requirements for processing grey-scale images; however, for motion, processing requirements would be much larger. Even though motion may be done entirely in grey-scale, it requires the storage and processing of numerous "time-slices" of a scene. Without some method to reduce the complexity, the number of calculations would grow by the number of time-slices required, as well as the number of trajectories which would be required. As a result, the processing of motion would need to be accomplished at some lower resolution, just as it appears to be done in the human visual system.

Another area of attention which requires further study is an investigation of search strategies. There appear to be many different possible strategies. Observers seem to be able to pick from among these and even to use more than one strategy in a particular search. Apparently, an

observer begins his search with a provisional strategy selected in view of the context of the overall situation. This context includes prompts, training, experience, and other factors. Once the observer has located a potential feature in the scene he then adopts a search strategy appropriate to that stimulus. One such possible strategy would be to overlay a model of the object which he expects is present, or for which he is searching, over the field of attentional indicators. The particular areas where the model and the attentional indicators coincide could then be examined for confirmation of the particular details expected.

5.2.2. Pseudo-neocognitron

The pseudo-neocognitron has the potential of providing a useful recognition structure. The benefits which it can potentially provide - the ability to recognize and reconstruct a distorted image - are unique. The significance of the reconstruction is that the object is reconstructed in its distorted state. This is important because components which make up the object maintain their original relationships instead of being forced into a pristine model. In the case of a weapon system, the difference between using a pristine representation and using the actual distorted model can mean the difference between success and failure. If the particular angle of a tank is such that it remains recognizable, but somewhat shorter in appearance than normal, a strike toward the engine compartment may miss completely in the case of a pristine representation.

As it currently stands, the pseudo-neocognitron as developed here uses Gabor wavelets throughout its computational structure. While this provides some degree of flexibility, the overall system performance may be better served by using some more conventional type of structure in its upper levels. Another of the drawbacks to the pseudo-neocognitron model is the large number of computations required to accomplish its tasks.

Although there are particular problems with the pseudo-neocognitron, the basic format of the neocognitron structure is significant, both in its recognition potential and in the ways in which

it seems to mimic natural functions. The structure of working from small localized functions replicated across the field of view to larger-scale more specific recognition functions, which then generate a representation of the object in its distorted appearance, appears to duplicate some hypothesized patterns in the brain. In the brain this begins with the Gabor-like structures in the striate cortex, and advances into the associative cortex. The oscillations inherent in this type of a model have also attracted the attention of brain researchers [Gray and Singer; Stryker].

5.2.3. Reverse Engineering

The AFIT Reverse Engineering System (ARES) has proven to be a useful concept; however, it has reached the stage where it requires a concerted effort to bring the pieces out of the laboratory and combine them into a coherent system. This effort is not suitable for a typical AFIT thesis project as the breadth of the effort required is too large for any particular research area, and yet not technical enough to be considered as doctoral-level work. Any student who began this task would find himself only beginning to learn the required tools by the time he would be expected to produce a thesis. A repeating cycle of students who never quite achieve the levels required in time to effect any real solutions could develop. As a result, the best solution may be to involve personnel whose sole duty is work on this aspect of the problem, with students doing research on particular portions of the system.

The integration of the portions of ARES into a coherent unit will require the generalization of the control section to include all of the researched processes. In several cases, selections from several candidate processes must be made. The choices depend on the particular circuit undergoing reverse engineering, the structure of a particular area on that chip, and the history of past processing in that area. This is the type of decision making which is well suited to heuristic processing methods. However, there are also large numbers of sequential steps which need to be accomplished between decisions. As a result, standard rule-based systems are not particularly well

suited for the task as a whole. Nor are conventional rule-based systems well suited for the initial stages of reverse-engineering a circuit, when the system must be trained for the particulars of the circuit. This training session requires a great deal of interaction with the system user. At the same time, the system must search through its own knowledge of how to reverse-engineer circuits to find the proper techniques to apply to the circuit. Much of this effort also requires long sequences of processes. The control system and the initial training of the reverse-engineering system could both benefit from the introduction of some object-oriented methods, possibly in combination with a rule-based system such as CLIPS. It is possible that the new object oriented extensions to CLIPS could fulfill this function. Another alternative may be to use C++. Basic research into the possible benefits of either choice and the initial application would be within the scope of a thesis project.

There is still more work which needs to be completed in incorporating improved techniques for the segmentation of areas of contiguous common composition on circuits. Although basic techniques have been constructed for the location of edges and to segment the scene, it is likely that no single technique will resolve these regions, or the edges of these regions, to the degree required. It is more likely that the best result will come through the combination of several of these techniques into some type of improved segmentation system. Further study is recommended into combining Gabor-gradient-direction methods with a region growing system which also considers edge strength and other characteristics.

Currently ARES uses a system of optical sensors. These sensors are adequate for the current generations of VLSI circuits. However, as the feature sizes continue to shrink, the sensors will reach the limits of their capabilities. Additionally, it is interesting to explore the capabilities of other sensors just to see what new types of insights they can offer. One of the more fascinating possibilities is the scanning electron microscope. New techniques for using this instrument allow the non-destructive investigation of VLSI circuits, possibly even under load conditions.

ARES also requires improvements in a number of areas which have not been discussed in detail in this document. Among these is the circuit builder, an expert system which reasons about the information gathered and infers information about portions of the circuit not visible in surface investigations. This subsection of the system could benefit from a system for reasoning under uncertainty.

5.2.4. Use of the CHIP System

The CHIP system has potential far beyond ARES. It is a general image processing system, with CAD and Expert System capabilities. This system could be put to work in a variety of tasks which require either image processing, or some combination of the capabilities of the system. A typical application might involve a silicon compilation system which uses an expert system to guide the design and placement of electronic components. The involvement of the network simulator means that CHIP could be used for a project that trains a neural net to recognize certain objects, and then produces the schematics for a VLSI circuit to implement that recognition. The possibilities are vast.

In order to realize the possibilities inherent in CHIP, there are a number of improvements which need to be made to the system. Currently, the system is implemented at a research level. This means that the routines do not include a large range of error checking, nor are they consistent in their application. The system assumes a large degree of a priori knowledge on the part of the user. As a result, CHIP can be somewhat difficult to learn and use. This could be improved through an effort to build a production-quality implementation of the system. The effort to create such an implementation of the system could also be used to free the system of any machine dependencies, to add a help system and to add commonly used functions which, for lack of want, have never been written. The display of a machine-independent version of CHIP could be via X-windows.

CHIP could also benefit from a variety of other upgrades and improvements. The current menu system for image-processing functions could be replaced with a lexical analyzer, allowing a more flexible and descriptive grammar. This would add to the ease of use of the system, and would allow easier expansion. The portions of the system, i.e., MAGIC, CLIPS, and CHIP, could be divided into a number of independent processes communicating through pipes. This would reduce the memory requirements for CHIP and would eliminate the long delays in accessing the other portions of CHIP while lengthy image-processing tasks are being performed.

Finally, CHIP should be brought under some sort of production control system. These systems (RCS, SCCS, etc.) provide a means of controlling the modifications to the sources. Some attempts have already been made in this area through the use of Makefiles and system-standardization efforts. However with the growth of CHIP to such a large system, the control of updates and modifications needs to be brought under a formal system.

References

- Adelson, Edward H. and James R. Bergen. "Spatiotemporal Energy Models for the Perception of Motion", *Journal of the Optical Society of America*, Vol 2. No 2. Pp. 284-299. (February 1985).
- Adelson, Edward H. and J. Anthony Movshon. "Phenomenal Coherence of Moving Visual Patterns", *Nature*, Vol 300. Pp. 523-525. (December 1982).
- Arbib, Michael A. "Vision, Brain, and Cooperative Computation: An Overview", *Vision, Brain, and Cooperative Computation*. ed. Michael A. Arbib and Allen R. Hanson. Cambridge, MA: The MIT Press. Pp. 1-83. (1990).
- Arbib, Michael A. and Donald H. House. "Depth and Detours: An Essay on Visually Guided Behavior", *Vision, Brain, and Cooperative Computation*. ed. Michael A. Arbib and Allen R. Hanson. Cambridge, MA: The MIT Press. Pp. 129-164. (1990).
- Augustus, Eric P. "VLSI Circuit Layer Determination by Reflectance for Use in Reverse Engineering." *MS Thesis, AFIT/GEO/ENG/90D-01*. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH. (December 1990).
- Aviation Week and Space Technology*. P. 77 (February 13, 1989).
- Bailey, Jim. and Dan Hammerstrom. "Why VLSI Implementations of Associative VLCNs Require Connection Multiplexing" in *IEEE International Conference on Neural Networks*. Vol. 2. Pp. 173-180. (1988).
- Baker, C. M. and C. Terman, "Tools for Verifying Integrated Circuit Designs", *Lambda*, Vol 1. No 3. Pp. 22-30. (1980).
- Barlow, Robert B. "What the Brain Tells the Eye", *Scientific American*. Pp. 90-94. (April 1990).
- Bedrosian, S.D. "The Role of Pattern Recognition in VLSI Testing," *1986 IEEE International Test Conference*. Pp. 750-754 (September 1986).
- Bovik, Alan C., Marianna Clark and Wilson S. Geisler. "Multichannel Texture Analysis Using Localized Spatial Filters" in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 12. No 1. Pp. 55-73. (January, 1990).
- Burns, J. Brian, Allen R. Hanson, and Edward M. Riseman. "Extracting Straight Lines", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol PAMI-8. No 4. Pp. 425-455. (July 1986).
- Burt, P. "Fast Algorithms for Estimating Local Image Properties", *Computer Vision, Graphics and Image Processing*, Vol 21. Pp. 368-382. (1983).
- Carlton, Eloise H. "Connection Between Internal Representation of Rigid Transformation and Cortical Activity Paths", *Biological Cybernetics*, Vol 59. Pp. 419-429. (1988).

- Chamberlin, Gordon J. and D. G. Chamberlin. *Colour: Its Measurement, Computation and Application*. Philadelphia: Heyden. (1980).
- Churchland, P.M. "Eliminative Materialism and the Propositional Attitudes" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 206-223. (1990).
- Churchland, P.M. and P.S. Churchland. "Stalking the Wild Epistemic Engine" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 300-311. (1990).
- Churchland, P.S. and Terrence J. Sejnowski. "Neural Representation and Neural Computation" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 224-252. (1990).
- Citron, Mark C., Robert C. Emerson, and William R. Levick. "Nonlinear Measurement and Classification of Receptive Fields in Cat Retinal Ganglion Cells", *Annals of Biomedical Engineering*, Vol 16. Pp. 65-77. (1988).
- Clark, Marianna and Alan C. Bovik, "Texture Discrimination Using A Model of Visual Cortex", *Proceedings of the IEEE International Meeting on Cybernetics and Society*. Pp. 1425-1430. (1986).
- Colthup, Norman B., Lawrence H. Daly, and Stephen E. Wiberley. *Introduction to Infrared and Raman Spectroscopy*. New York: Academic Press. (1975).
- Curie, D. "Luminescent Processes in Semiconductors", *Luminescence of Inorganic Solids*. New York: Plenum. (1978).
- Daugman, John G., "Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36. Pp. 1169-1179. (July 1988).
- Daugman, John G., "Two Dimensional Analysis of Cortical Receptive Field Profiles", *Vision Research* Vol 20. Pp. 846-856. (1980).
- Daugman, John G., "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters", *Journal of the Optical Society of America*, Vol 2. Pp. 1160-1169. (July 1985).
- Dennett, Daniel C. "Making Sense of Ourselves" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 184-198. (1990a).
- Dennett, Daniel C. "True Believers" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 150-166. (1990b).
- Dennett, Daniel C. "Why the Law of Effect Will Not Go Away" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 63-77. (1990c).
- Descartes, Rene. *Rules for the Direction of the Mind. Discourse on the Method*. New York: Encyclopedia Britannica. (1955).
- Diamond, Marian C., Arnold B. Scheibel and Lawrence M. Elson. *The Human Brain Coloring Book*. New York: Barnes and Noble. (1985).
- DiGiacomo, Joseph. *VLSI Handbook*. New York: McGraw-Hill Publishing Company. (1989).

- Dukes, Michael A. "A Multiple-Valued Logic System for Circuit Extraction to VHDL 1076-1987." *MS Thesis, AFIT/GE/ENG/88S-1*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (September 1988).
- Ekström, P. and H. Meissl. "Signal Processing in a Simple Vertebrate Photoreceptor System -- The Teleost Pineal Organ", *Physiologia Bohemoslovaca*, Vol 38. No 4. Pp. 311-326. (1989).
- Emerson, Robert C. *Telephone Conversation*. (August 1990).
- Emerson, Robert C., James R. Bergen, and Edward H. Adelson. "Directionally Sensitive Complex Cells and the Computation of Motion Energy in Cat Visual Cortex" submitted to *Vision Research*. (March 1990).
- Ervin, Frank R. "On-line Computer Techniques for Analysis of the Visual System", *Symposium on the Analysis of Central Nervous System and Cardiovascular Data Using Computer Methods*. Washington, D. C. (October 1964).
- Evans, Ralph M. *An Introduction to Color*. New York: John Wiley & Sons. (1948).
- Findlay and Crawford, "Limited Plasticity of Saccades" in *Eye Movements and Psychological Processes*. Ed. Rudolf Groner et al. Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 191-212. (1983).
- Fiorentini, Adriana. et al. "The Perception of Brightness and Darkness: Relations to Neuronal Receptive Fields" in *Visual Perception: The Neurophysiological Foundations*. Ed. Lothar Spillmann and John S. Werner. New York: Academic Press. Pp. 129-162. (1990).
- Fistul', Victor I. *Heavily Doped Semiconductors*. Trans. Albin Tybulewicz. New York: Plenum Press. (1969).
- Ford, A. C.T. White and M. Lichtenstein. "Analysis of Eye Movement during Free Search" in *Journal of the Optical Society of America*. Vol. 49. Pp. 165-177. (1959).
- Fretheim, Erik J. "Reverse Engineering VLSI Using Pattern Recognition Techniques." *MS Thesis, AFIT/GE/ENG/88J-1*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (June 1988).
- Fretheim, Erik J. and Matthew Kabrisky. "The Use of the Power Cepstrum in the Analysis of VLSI Circuits" presented at NAECON '90. (April 1990).
- Fukushima, Kunihiro. "Neural Network Model for Selective Attention in Visual Pattern Recognition and Associative Recall" in *Journal of Applied Optics*. Vol. 26. No. 23. Pp. 4985-4992. (December 1987).
- Fukushima, Kunihiro and Sei Miyake. "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position" in *Pattern Recognition*. Vol. 15. No. 6. Pp. 455-469. (1982).
- Gabor, Dennis, "Theory of Communication", *Journal of IEE*, Vol 93. Pp. 429-457. (1946).
- Geiger, Randall L., Phillip E. Allen, and Noel R. Strader. *VLSI Design Techniques for Analog and Digital Circuits*. New York: McGraw-Hill Publishing Company. (1990).

- Giarratano, Joseph C. *CLIPS User's Guide; Version 4.3 of CLIPS*. Lyndon B Johnson Space Center, Tx. (August 1989).
- Ginsburg, Arthur P. "Spatial Filtering and Visual Form Perception" in *Handbook of Perception and Human Performance*. Vol II. Ed. Kenneth R. Boff et al. New York: Wiley & Sons. Section 34. (1986).
- Goldstein, E. Bruce. *Sensation and Perception, 3rd Ed.* Belmont, California: Wadsworth Publishing Company. (1989).
- Gould, John D. "Eye Movements During Visual Search", *Proceedings of 1969 NATO Symposium on Image Evaluation*. ed. H. W. Leibowitz. Munich, Germany. Pp. 145-160. (1969).
- Gould, John D. "Eye Movements During Visual Search and Memory Search", *Journal of Experimental Psychology*. Vol 98. No 1. Pp. 184-195. (1973).
- Gray, Charles M. and Wolf Singer. "Stimulus-specific Neuronal Oscillations in Orientation Columns of Cat Visual Cortex", *Proceedings of the National Academy of Science, USA*, Vol. 86. Pp. 1698-1702. (March 1989).
- Green, William B. *Digital Image Processing*. New York: Van Nostrand Reinhold Company. (1983).
- Hardy, Arthur C. et al. *Handbook of Colorimetry*. Cambridge: The Technology Press. (1936).
- Harter, M. Russel and Cheryl J. Aine. "Brain Mechanisms of Visual Selective Attention", *Varieties of Attention*. ed. Raja Parasuraman. New York: Academic Press. Pp. 293-321.(1984).
- Hayden, Richard A. "Analysis System for Reverse Engineering VLSI Circuits." *MS Thesis, AFIT/GE/ENG/89D-17*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1989).
- Heavens, O.S. *Optical Properties of Thin Solid Films*. New York: Academic Press. (1955).
- Heeger, David J. "Optical Flow from Spatiotemporal Filters", *IEEE Proceedings of the International Conference on Computer Vision (ICCV)*. Pp. 181-190. (June 1987).
- Hines, Willian W. and Douglas C. Montgomery. *Probability and Statistics in Engineering and Management Science*. 2nd Ed. New York: Wiley & Sons. (1980).
- HMOF Facility Description*. Internal Memorandum. Helmet Mounted Oculometer Facility, Armstrong Medical Research Laboratory. Wright-Patterson AFB OH.
- Holten, James R. "A Robot Vision System," *PhD Dissertation, AFIT/DS/ENG/85D-1*, School of Engineering, Air Force Institute of Technology (AU). (December 1985).
- Holtzman, Jeffrey, Bruce T. Volpe and Michael S. Gazzaniga. "Spatial Orientation Following Commissural Section", *Varieties of Attention*. ed. Raja Parasuraman. New York: Academic Press. Pp. 376-379.(1984).
- Hubel, D.H. and T.N. Wiesel. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex" in *Journal of Physiology*. Vol. 160. Pp. 106-154. (1962).

- Hubel, David H. and Torsten N. Wiesel. "Sequence Regularity and Geometry of Orientation Columns in the Monkey Striate Cortex" in *Computational Neurology*. Vol. 158. Pp. 267-294.
- Jackson, Frank. "Epiphenomenal Qualia" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 469-477. (1990).
- Jones, Judson P., Aaron Stepnoski and Larry Palmer. "The Two-Dimensional Spectral Structure of Simple Receptive Fields in Cat Striate Cortex" in *Journal of Neurophysiology*. Vol. 58. No. 6. Pp. 1212-1232. (December 1987).
- Jones, Judson P. and Larry Palmer. "The Two-Dimensional Spatial Structure of Simple Receptive Fields in Cat Striate Cortex" in *Journal of Neurophysiology*. Vol. 58. No. 6. Pp. 1187-1211. (December 1987).
- Jones, Judson P. and Larry A. Palmer, "An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex", *Journal of Neurophysiology*, Vol. 58. Pp. 1233-1258. (December 1987).
- Kabriskey, Matthew. "Feathers and Birds, Neurons and Computers." Unpublished. (1967).
- Kabriskey, M. et al. "A Theory of Pattern Perception Based on Human Physiology", *Contemporary Problems in Perception*. ed. A. T. Welford and L. H. Housiadas. London: Taylor and Francis. Pp. 129-147. (1970).
- Kabriskey, Matthew. *A Proposed Model for Visual Information Processing in the Brain*. Urbana: University of Illinois Press. (1966).
- Kapoula, Zoi. "The Influence of Peripheral Preprocessing on Oculomotor Programming in a Scanning Task", *Eye Movements and Psychological Functions: International Views*. ed. Rudolf Groner et al. Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 101-114. (1983).
- Karsh, Robert and Francis W. Breitenbach. "Looking at Looking: The Amorphous Fixation Measure" in *Eye Movements and Psychological Functions: International Views*. ed. Rudolf Groner et al. Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 53-64. (1983).
- Kedem, G. "The Quad-CIF Tree: A Data Structure for Hierarchical On-line Algorithms", *Proceedings of the International Conference on VLSI*. The Netherlands: North Holland. Pp. 352-357. (1983).
- Lambert, Laurence. "Evaluation and Enhancement of the AFIT Autonomous Face Recognition Machine", *Thesis AFIT/GE/ENG/87D-35* School of Engineering, Air Force Institute of Technology (AU). (December 1987).
- Land, E. H. "Recent Advances in Retinex Theory and Some Implications for Cortical Computations: Color Vision and the Natural Image", *Proceedings of the National Academy of Science, USA*. Vol 80. Pp. 5163-5169. (1983).
- Larson, Gary. *Farside*. Universal Syndicates. (1989).
- Leano, Gene E. "Reverse Engineering VLSI Using Lasers and Laser-like Light Sources." *MS Thesis, AFIT/GE/ENG/89D-6*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1989).
- Lee, C. Y. "An Algorithm for Path Connections and Its Applications", *IRE Transactions on Electronic Computers*. Pp. 346-365. (1961).

- Luria, Aleksandr R. *Higher Cortical Functions in Man*. New York: Basic Books, Inc. (1966).
- Lycan, William G. ed. *Mind and Cognition: A Reader*. Guildford, Great Britain: Biddles Ltd. (1990).
- Mackworth, Norman H. "Stimulus Density Limits the Useful Field of View" in *Eye Movements and Psychological Processes*. Ed. Richard Monty and John Senders. Hillsdale, NJ: Lawrence Erlbaum Associates. Pp. 305-321. (1976).
- Mallat, Stephane G. *Complete Signal Representation With Multiscale Edges. Technical Report No. 483*. Department of Computer Science, New York University. New York. (December 1989a).
- Mallat, Stephane G. "Multifrequency Channel Decomposition of Images and Wavelet Models", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol 37. No 12. Pp. 2091-2110. (December 1989b).
- Mallory, W. *Eye Safety for Honeywell Oculometer. Systems Research Laboratory Technical Memorandum 5020-44-M-85-7-1*. (July 1981).
- Marcelja, S. "Mathematical Description of the Responses of Simple Cortical Cells" in *Journal of the Optical Society of America*. Vol. 70. Pp. 1297-1267. (1980).
- Marr, David. *Vision*. New York: W. H. Freeman and Company. (1982).
- Masland, Richard H. "The Functional Architecture of the Retina", *Scientific American*. Pp. 102-111. (December 1986).
- McConkie, George W. "Evaluating and Reporting Data Quality in Eye Movement Research", *Behavior Research Methods and Instrumentation*. Vol 13. No 2. Pp. 97-105. (1981).
- Miller, K. "Call to the Colors: Progress Toward a Three-dimensional Standard", *Photonics Spectra*, Vol 19. No 2. Pp. 75-82. (1985).
- Mocharnuk, John B. "Visual Target Acquisition and Ocular Scanning Performance", *Human Factors*. Vol 20. No 5. Pp. 611-631. (1978).
- Moon, Parry. *Scientific Basis of Illuminating Engineering*. New York: McGraw-Hill. (1936).
- Moss, T.S. *Optical Properties of Semi-conductors*. London: Butterworths Scientific Publications. (1959).
- Mueller, Michael R. "Investigation of Gabor Filters for Use in Reverse Engineering VLSI Circuits." *MS Thesis, AFIT/GE/ENG/89D-35*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1989).
- Mueller, Michael R. et al. "Gabor Transforms to Preprocess Video Images for Back-Propagation" presented at *NAECON '90*. (1990).
- Nelson, Ralph, and Helga Kolb. "Synaptic Patterns and Response Properties of Bipolar and Ganglion Cells in the Cat Retina", *Vision Research*, Vol 23. No 10. Pp. 1183-1195. (1983).
- Newell, M. E. and D. T. Fitzpatrick. "Exploiting Structure in Integrated Circuit Design Analysis", *Proceedings of the Conference on Advanced Research in VLSI*. Massachusetts Institute of Technology: Cambridge, Massachusetts. (January 1982).

- Nussbaum, Allen and Richard A. Phillips. *Contemporary Optics for Scientists and Engineers*. New York: Prentice-Hall. (1976).
- Ousterhout, John K. "The User Interface and Implementation of an IC Layout Editor", *IEEE Transactions on Computer-Aided Design, Vol CAD-3. No 3*. Pp. 242-249. (July 1984).
- Ousterhout, John K. et al. "Magic: A VLSI Layout System", *Proceedings of the 21st Design Automation Conference*. Pp. 152-159. (1984).
- Ousterhout, John K. "Corner Stitching: A Data Structuring Technique for VLSI Layout Tools", *IEEE Transactions on CAD/ICAS, Vol CAD-3. No 1*. Pp. 87-99. (1984).
- Parthasarathy, Srinivasan. et al. "A Color Vision System for Film Thickness Determination," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation. Vol I*. Pp. 515-519 (June 1987).
- Pavlidis, Theo. *Algorithms for Graphics and Image Processing*. Rockville, Md: Computer Science Press. (1982).
- Pavlidis, Theo and Yuh-Tay Liow. "Integrating Region Growing and Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12. No 3*. Pp. 225-233. (March 1990).
- Penrose, Roger. *The Emperor's New Mind*. New York: Oxford University Press. (1990).
- Place, U.T. "Is Consciousness a Brain Process?" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 29-36. (1990).
- Pollen, Daniel A. and Steven F. Ronner. "Phase Relationships Between Adjacent Simple Cells in the Visual Cortex" in *Science*. Vol. 212. Pp. 1409-1411. (June 1981).
- Polyak, S. *The Vertebrate Visual System*. Ed. H. Kluver. Chicago: University of Chicago Press. (1957).
- Powell, R. C. "Modern Techniques in Optical Spectroscopy", *Luminescence of Inorganic Solids*. New York: Plenum. (1978).
- Prinz, Wolfgang. "Asymmetrical Control Areas in Continuous Visual Search", *Theoretical and Applied Aspects of Eye Movement Research*. ed. A. G. Gale and F. Johnson. New York: Elsevier Science Publishers. Pp. 85-100. (1984).
- Querns, John F. "Segmentation of Regions of Contiguous Common Composition on VLSI Circuits." *MS Thesis, AFIT/GEO/ENG/9D-08*. Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1990).
- Rabbitt, Patrick. "The Control of Attention in Visual Search", *Varieties of Attention*. ed. Raja Parasuraman. New York: Academic Press. Pp. 273-291.(1984).
- Randall, R.B. *Application of B & K Equipment to Frequency Analysis*. Naerum, Denmark: Bruel and Kjaer Instruments Inc. (1977).
- "Recommendations on Uniform Color Spaces - Color Difference Equations, Psychometric Color Terms." *Supplement No. 2 to CIE Publication No. 15 (E-1.3.1)1971(TC-1.3)*. Bureau Central de La CIE, Paris. (1978).

- Roberts, Richard. "Three-Dimensional Scene Analysis Using Stereo Based Imaging," *MS Thesis, AFIT/GE/ENG/87D-54*, School of Engineering, Air Force Institute of Technology (AU). (December 1987).
- Robinson, Gordon H., Bruce W. Koth and John P. Ringenbach. "Dynamics of Eye and Head During an Element of Visual Search", *Ergonomics*. Vol. 19. No. 6. Pp. 691-709. (1976).
- Rock, I. "The Description and Analysis of Object and Event Perception" in *Handbook of Perception and Human Performance*. Vol II. Ed. Kenneth R. Boff et al. New York: Wiley & Sons. Section 33. (1986).
- Rodieck, R. W. and I. Stone. "Analysis of Receptive Field of Cat Retinal Ganglion Cells", *Journal of Neurophysiology*, Vol 28. Pp. 833-849. (1965).
- Schumacher, W. and A. Korn. "Automatic Evaluation of Eye or Head Movements for Visual Information Selection", *Theoretical and Applied Aspects of Eye Movement Research*. ed. A. G. Gale and F. Johnson. New York: Elsevier Science Publishers. Pp. 31-42. (1984).
- Scott, Walter S. and John K. Ousterhout. "Plowing: Interactive Stretching and Compaction in Magic", *Proceedings of the 21st Design Automation Conference*. Pp. 160-165. (1984).
- Scott, Walter S. and John K. Ousterhout. "Magic's Circuit Extractor", *Proceedings of the 21st Design Automation Conference*. Pp. 160-165. (1984).
- Shapley, Robert. et al. "Computational Theories of Visual Perception" in *Visual Perception: The Neurophysiological Foundations*. Ed. Lothar Spillmann and John S. Werner. New York: Academic Press. Pp. 417-448. (1990).
- Shoop, Michael R. "An Expert System to Aid the Reverse Engineering of Integrated Circuits." *MS Thesis, AFIT/GE/ENG/90D-6*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1990).
- Snyder, Harry L. and Donald F Taylor. *Computerized Analysis of Eye Movements During Static Display Visual Search*. AMRL-TR-75-91. Aerospace Medical Research Facility: Wright-Patterson AFB. (February 1976).
- Sober, Elliott. "Putting the Function Back into Functionalism" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 97-106. (1990).
- Sparks, David L. and Martha Jay. "The Role of the Primate Superior Colliculus in Sensorimotor Integration", *Vision, Brain, and Cooperative Computation*. ed. Michael A. Arbib and Allen R. Hanson. Cambridge, MA: The MIT Press. Pp. 109-128. (1990).
- Sperling, George. "Attention and Signal Detection", *Varieties of Attention*. ed. Raja Parasuraman. New York: Academic Press Pp. 105-113.(1984).
- Spinelli, D. N. "A Trace of Memory: An Evolutionary Perspective on the Visual System", *Vision, Brain, and Cooperative Computation*. ed. Michael A. Arbib and Allen R. Hanson. Cambridge, MA: The MIT Press. Pp. 165-182. (1990).
- Sterling, Peter. "Microcircuitry of the Cat Retina", *Annual Review of Neuroscience*, Vol 6. Pp. 149-185. (1983).
- Sterling, Peter, Michael A. Freed, and Robert G. Smith. "Architecture of Rod and Cone Circuits to the On-beta Ganglion Cell", *The Journal of Neuroscience*, Vol 8. No 2. Pp. 623-642. (February, 1983).

- Stryker, Michael P. "Is Grandmother an Oscillation?" *Nature*, Vol. 338. p297. (March 1989).
- Suter, Bruce W. and Matthew Kabrisky. "On a Magnitude Preserving Iterative MAXnet Algorithm" submitted to *Journal of Neural Computation*. (April 1990).
- Tauchi, Masaki and Richard H. Masland. "Local Order among the Dendrites of an Amacrine Cell Population", *The Journal of Neuroscience*, Vol 5. No 9. Pp. 2494-2501. (September 1985).
- Taylor, George S. and John K. Ousterhout. "Magic's Incremental Design-Rule Checker", *Proceedings of the 21st Design Automation Conference*. Pp. 160-165. (1984).
- Treisman, Anne. "Preattentive Processing in Vision" in *Computer Vision, Graphics and Image Processing*. Vol. 31. Pp. 156-177. (1985).
- Treisman, Anne. "Properties, Parts and Objects" in *Handbook of Perception and Human Performance*. Vol II. Ed. Kenneth R. Boff et al. New York: Wiley & Sons. Section 35. (1986).
- Treisman, Anne et al. "Form Perception and Attention: Striate Cortex and Beyond" in *Visual Perception: The Neurophysiological Foundations*. Ed. Lothar Spillmann and John S. Werner. New York: Academic Press. Pp. 273-316. (1990).
- Turner, M.R., "Texture Discrimination by Gabor Functions", *Biological Cybernetics*, Vol. 55. Pp. 71-82. (1986).
- Van Gulick, Robert. "Functionalism, Information and Content" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 107-129. (1990).
- Vardi, N., P.J. Masarach, P. Sterling. "Structure of the Starburst Amacrine Network in the Cat Retina and Its Association with Alpha-Ganglion Cells", *Journal of Comparative Neurology*, Vol 288, No 4. Pp. 601-611 (1989).
- Walpole, Ronald E. and Raymond H. Myers. *Probability and Statistics for Engineers and Scientists*. 2nd Ed. New York: Macmillan Publishing Company. (1978).
- Watson, A. B., and A. J. Ahumada. "Model of Human Visual Motion Sensing", *Journal of the Optical Society of America, A*. Vol 2. No 2. Pp. 322-342. (1985)
- Watson, J. B. "An Excerpt from 'Talking and Thinking' Chapter 10 of *Behaviourism*" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 14-22. (1990).
- Werblin, Frank S. "The Control of Sensitivity in the Retina", *Scientific American*, Vol 228. Pp. 70-79 (January 1973).
- Weste, N. H. E. and K. Eshraghian. *Principles of CMOS VLSI Design*. Reading PA: Addison-Wesley Publishing Company. (1985).
- Widdel, Heino. "Operational Problems in Analysing Eye Movements", *Theoretical and Applied Aspects of Eye Movement Research*. ed. A. G. Gale and F. Johnson. New York: Elsevier Science Publishers: Pp. 21-29. (1984).

- Wolff, Lawrence B. and Terrance E. Boulton. "Polarization/Radiometric Based Material Classification" in *Proceedings: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington: IEEE Computer Society Press. Pp. 387-395. (June 1989).
- Wright, W.D. *The Measurement of Colour*. New York: D. Van Nostrand. (1964).
- Wurtz, Robert H., Michael E. Goldberg and David Lee Robinson. "Brain Mechanisms of Visual Attention", *Scientific American* Pp. 124-135.
- Yarbus, Alfred L. *Eye Movements and Vision*. New York: Plenum Press. (1967).
- Zrenner, Eberhart, et al. "Color Perception: Retina to Cortex" in *Visual Perception: The Neurophysiological Foundations*. Ed. Lothar Spillmann and John S. Werner. New York: Academic Press. Pp. 163-204. (1990).

Rock, I. "The Description and Analysis of Object and Event Perception" in *Handbook of Perception and Human Performance. Vol II.* Ed. Kenneth R. Boff et al. New York: Wiley & Sons. Section 33. (1986).

APPENDIX A

Vision

1. Introduction

It is perhaps in the realization of the generalized vision system that we will begin to approach the ultimate goal of artificial intelligence, the ability to reason. For in biology it was the ever-increasing efforts to exploit the potential of the visual world which, along with the other demands of evolution, extended the size of the brain. Exploiting capabilities inherent in vision may prove helpful to the strong AI community, those who believe that it is possible to create a machine which gains, or appears to have gained, consciousness. Current approaches to AI have been logic or knowledge based. They have been built around attempts to acquire and organize knowledge, or upon logic systems and logical manipulations of logic. These systems have not yet been able to provide an ultimate thinking machine or even meet the early promises of AI. The problem with these machines is that their very premise is probably flawed. These systems ignore the paradigm of nature in which intelligence, learning and consciousness developed to their fullest extent in visually-oriented systems. Rather than being the basis of intelligence, logic and knowledge are tools and products to support a vision-based intelligence. Block, for whom the term analog¹ is equivalent to pictorial, states, "the real danger for artificial intelligence is that the model might soon become an unimportant digital computer coupled to an important analog computer. [Block, p. 599]." This model is accepted unconsciously by our society. We speak of

¹Analog in the sense referred to by Block is not the sense in which we normally use it as engineers, but rather analog in the sense that the results of a process are "lawfully dependent on the character of the input [Block, p. 605]." A picture is analog in that "[p]ictorial representation involves analog representation of the spatial properties of the situation presented [Sterelny, p 613]." This does not preclude its representation in a digital form, but rather demands only that the representation be *lawfully* related to the input. Block, not being an engineer, uses *lawful* where we would normally expect to find the term *linear*.

"seeing" the essence of a proof, or of our "view" of the world. Thus, if we are to hope to create intelligence we must "seek" an understanding of vision.

This natural connection between our "folk philosophic" view of how our intelligence operates and our vision, needs to be exploited. This can be done by using vision techniques for tasks that have been relegated to logic systems. The consolidation of letters into words for a reading machine is one such task. Many schemes have been devised to use semantic nets, production rules, etc., to form complete words out of collections of letters received from some "lower level" recognizer, and yet the general reading machine has remained elusive. However, O'Hair, by using pattern recognition techniques on entire words, has shown a phenomenal success for this task [O'Hair]. Dreyfus and Dreyfus contended that "pattern recognition may figure in even what seemed to be exemplars of high-level reasoning tasks that seemed to require rule-based reasoning [Bechtel, p. 263]." A specific example they considered was that of the chess player who becomes an expert not by knowing the rules better and analyzing the moves farther ahead, but rather through recognizing how the current setup of the board resembles a past one and applying knowledge of that past event to the current game [Bechtel]. Simon and Chase found that a world-class chess player will have memorized about 50,000 chess patterns, an experienced amateur about 1,000, and a novice none. The same exponential rises in memorized patterns could also be found in studies of expert players of go, gomoku, and bridge. In yet another study, Egan and Schwartz found that skilled electronics technicians understood circuit diagrams by grouping the components into known patterns - amplifiers, filters, rectifiers, etc. [Chase].

An approach to intelligence through pattern recognition also offers opportunities to reach solutions to the halting [entscheidung] problem. The claim has been made that the reason machines will never reach the capacities of man is because they cannot solve the halting problem, but men do. In this view let us consider a function, possibly some type of fractal, which generates a pattern, a part of which appears as shown in Figure 1. If this function were fed to a stan-

standard AI type program, and the program were asked if the function reaches some limit, the program would likely choke on its own recursion. However, if we were to correlate the outputs of the function at a scale of one and at some random scales n and $n+1$, we could reasonably determine that the function does not reach a limit.

Likewise, the approach to vision itself should be visually based. When Homunculus - the illusion that there is a little person inside our head, looks out at the world, he does not throw labels on everything and file it away in neat file cabinets divided into folders of events, but rather he uses the incoming information to paint a world scene and to place objects in our world view. The objects in this world view exist and can evoke responses from us without having labels. It is only when labels are needed that they are applied. In a room with a chair, we would be aware of

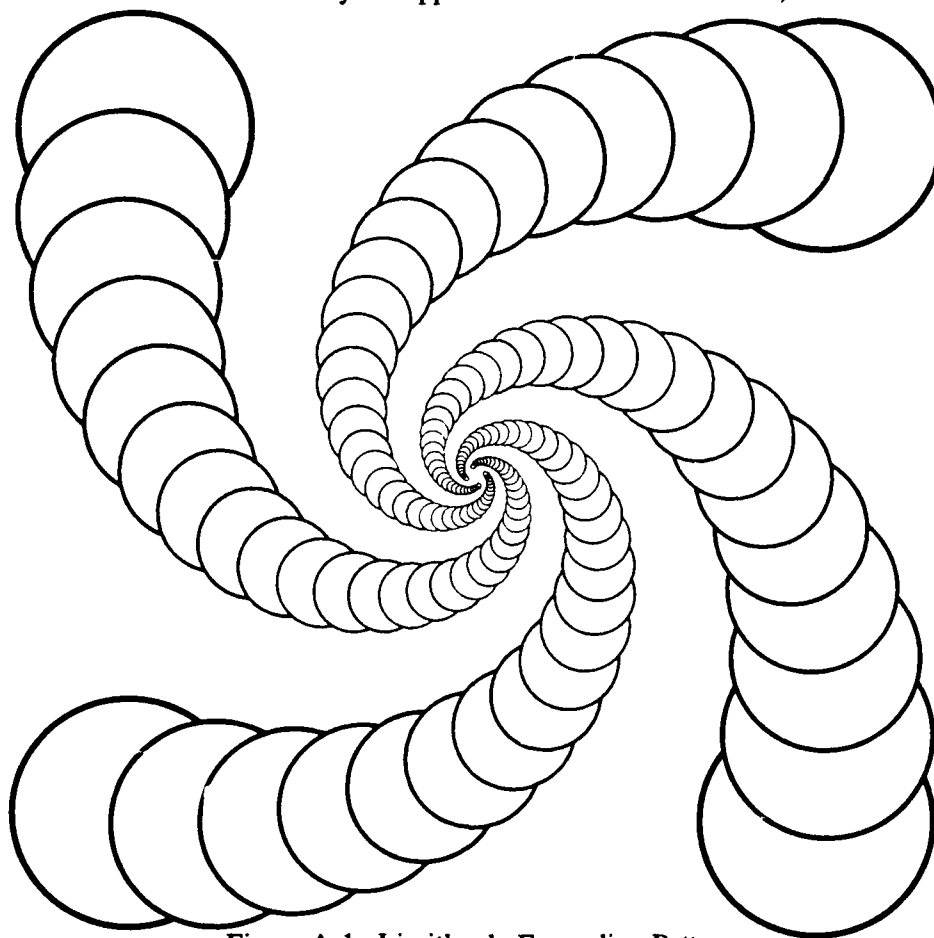


Figure A-1: Limitlessly Expanding Pattern

the existence of the chair and avoid bumping into it as we move about the room. We might also use it as a place to sit, but it is only when we attempt to explain the contents of the room to someone else that we apply the label of "chair" to the object. The world scene is not what our eyes are looking at, but rather the area which we perceive we are seeing. This is an illusionary area which despite the constant movement of our eyes appears unchanging before us. Our world view is our map of the world around us. The world view is the representation which allows us to maintain the relative positions of items about us even when we are not looking in their direction. An observer in a room standing facing a wall with a blackboard and a desk will see the desk in the world scene even as his eyes jump about on the blackboard and off to the sides. The observer's scene will not jump about even though his eyes jump about. As the observer begins to turn to the left, his world scene will begin to change, much in the same manner as a movie panning to the left. Meanwhile, the desk has been entered into his world view, not as a label, but as a known object which can be recognized as a desk when the need arises. Because the desk has been entered into his world view, he is not surprised by its reappearance when he returns to the right. If some other object were placed there while his back was turned he would be startled. The observer's surprise would be complete without his ever having labeled the desk as such, and without labeling the new object. In fact, the observer would be able to identify the switching of the two objects even if they were both unique, but totally unknown items. This feat is an accomplishment of visual processing and not data, knowledge or logic based processing.

Another area which shows the importance of a visual concept of processing is the relatively new field of *scientific visualization*. Many people are discovering that is easier to deal with and spot patterns in the massive amounts of data computers make available if the data are presented as visual patterns. The visual presentation allows people to jump outside of the data to get an insight into what the data are expressing. The idea of visualization of a problem is nothing new nor limited to being done on computers. People have long created "pictures" of abstract problems

in their minds in order to try to get a grasp of a difficult concept. People have also drawn pictures and developed diagramming methods to help them understand the nature of a purely mathematical numerical concept. This is the concept behind the use of a cartesian plane to illustrate (another visual word) complex numbers. The same concept could be employed by a computer to allow it to complete its own pattern recognition processes and obtain its own insight. Thus a technique designed to help people may also serve to provide computers with a valuable capability.

2. Defining Vision

The first of many daunting tasks in trying to develop a vision system is to define "vision". Part of the challenge of this task is that what constitutes vision is dependent on the particular environment in which it is being defined. However, in general we can say that vision is a process by which multidimensional spatial/spectral/temporal data are converted into a form which allows relevant action on the part of the possessor of the "vision" function.

This description is simple enough to cover the visual systems of lower animals which use the output of their vision systems to drive their reflexes, or to include the grocery store scanner which reads the UPC symbols from the products the customer has selected and outputs a product number which can then be used to obtain the price of the items, control inventories, perform ordering functions, or perform any one of a number of other tasks. This description is at the same time powerful enough to include a radar-based system used for navigating aircraft and locating potential targets, or to include the human visual system which presents to us the world in which we live, or at least our perception of it.

2.1. Seeing and Recognition

Having given a definition of "vision" we can try to understand what "seeing" and "recognition" are. "Seeing" and "recognition" are products of the vision process. When we "see" something we become aware of its presence. When we "recognize" something we become aware of what it is. That is, we produce a relevant relationship between that object and some other thing (objects, concepts, etc.). For example, in the case of a grocery scanner, it sees an item when it captures a product code from that item. The output of the scanner's vision system is a green or red light and a number. When the scanner blinks a red light at us it has seen an object, but it has not recognized that object. If it blinks a green light at us, the scanner is signalling that it has seen an object and established a relationship between it and some product code.

Sometimes "seeing" and "recognition" are not separable. An example maybe found in a frog; the frog's small moving-spot detector is at once a bug "seer" and a bug "recognizer". In this simple system, all small moving spots are classified the same so there is no difference between seeing and recognition. At other times the "seeing" may be dependent on the "recognition". An example presents itself in a Far Side cartoon (Figure 2). Here a deadly couch-snake has hidden itself in its natural environment. In order to become aware of the presence of the couch snake the inattentive gentleman must first recognize that it is indeed a couch snake and not a portion of the couch. He can then see the position of the cobra and take action to avoid it. This requirement for recognition before "seeing" is perhaps even more apparent in the photo "Pintos" (Figure 3). Both recognition and seeing are based on properties of the scene, the sensors and the vision system. It is the convergence of these which allows something to be seen and recognized. In order for this to happen one or more properties must be present in the scene which can be recorded by the sensor and segmented or distinguished by the vision system.



Figure A-2: Deadly Couch Cobra Awaits Its Victim [Larson]

2.2. Components of Vision

Vision has three basic components. The first of these, the scene, is the portion of the external world to which the vision system responds. The second is the sensor or sensors which translate some physical property of the external world into a signal internal to the vision system. The final portion is the production of relevant outputs. Each of these vision system components possesses a number of properties. The first set of properties includes those intrinsic to the scene being viewed. Examples of these are shape, motion, texture, and relations among objects in the scene. The second set of properties to be considered is those of the sensor or sensors being used. These would include such things as the operating frequencies, field of view, attention mechanisms, and focus. The third set of properties would include those of vision as a whole, such as the number of sensors and their interrelationships and how the system represents the scene internally.

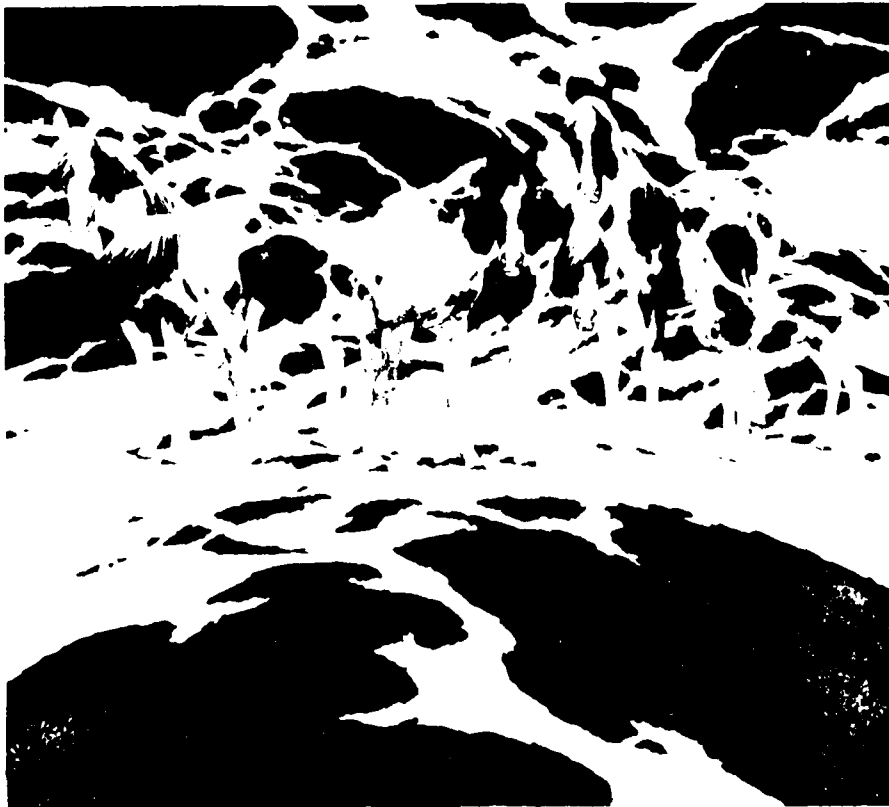


Figure A-3: *Pintos* by Bev Doolittle [Goldstien, p. 204]

There may appear to be a discontinuity when we discuss the third component of a vision system as the production of outputs, but the third set of properties as belonging to the vision system and its environment as a whole; however this is a direct result of the inseparability of the production of meaningful relevant outputs, and the environment and sensors of a vision system. Often, in fact, there is a large degree of feedback in the system, with this feedback the formation of output affects both the response of the sensors, and in some cases the environment of the scene itself. This kind of interplay becomes increasingly important as the complexity of the vision system grows.

The external environment which comprises the scene possesses a number of properties by which it can be characterized. It is the combinations of these properties which allows portions of

the scene to be "seen" and "recognized". These properties provide distinctions in space, frequency and time. Among these properties are shape, size, lighting, material composition, spatial relationships, texture, and reflectance. This short list does not span the possible space of properties, but rather is representative of the inherent properties of scenes which can prove useful to vision systems. Each of these properties has some inherent ambiguities associated with it. Shape can be used for differentiating between two masses, but a log viewed from the end has the same shape as a pie viewed from above. Likewise, the spectral reflectances of two objects may normally be very different, but when viewed under certain lighting conditions they may appear alike. Who has not staged a picture of someone resting his arm on a tall building or bridge? To overcome the inherent ambiguities, it is useful to involve as many scene properties as is practical in any classification attempt.

Properties of sensors can also be related to space, time and spatial frequency. The resolution of a sensor is a spatial property. It determines the amount of spatial information which can be obtained from any particular scene. Sensors are often viewed as providing limitations to the vision system. They have a limited spectral response; they possess a particular depth of field; or they cannot detect events of too short a duration. When viewed in this way, the sensors are an obstacle to be overcome by the rest of the system. However, sensors can also be viewed as filters which prevent the introduction of excessive, or worthless, information into the system. Seen in this light they capture only the pertinent portions of the spectrum; they disregard distant, unimportant objects; and they capture only events of a significant duration. The sensor is seen to be an important, integral piece of the overall system, not some limiting factor to be overcome. The system incorporates the bounds of the sensor into its design as useful capabilities.

Vision systems' limitations and capabilities are a result of the properties of the sensors they employ, the scenes and the internal workings of the particular vision systems. The properties of the vision system determine how it is going to take advantage of the information about the scenes

provided through the sensors. Vision-system properties can include such features as memory duration, abilities to exploit particular aspects of a scene, the speed with which the system can function, the system's reliability, and the generalizability of the system. The vision system can be highly tailored to a particular task, or it may be capable of performing a wide variety of tasks. Along the continuum is the ability to adapt the system to a particular task at any given moment and then to another task at a later time. The manner in which the system outputs the information which it has gathered from the scene is also an important element of the system's properties.

It is the combination of all of these properties - scene, sensor, and system - which determines whether a particular vision system is appropriate for a particular vision task. A system may have the desired outputs, the correct sensors, or some other appropriate property, but if it cannot function in the context of the scene, or more appropriately if the integrated properties of all three do not mesh together, the vision system will not be successful. Moreover, except in limited cases, the construction of a vision system should not be viewed as the development of a linear system with a straight path from input to output, nor should it be viewed as a closed-loop system where everything relevant is included in either the scene, the sensors, or the system. Instead, a vision system must be viewed as a much broader system subject to external pressures. To develop a system requires consideration of all of these elements and how they interact.

2.3. Two Kinds of Seeing

One further consideration in developing a vision system is the apparent existence of two kinds of "seeing" in the human visual system. These can be categorized as instantaneous and analytical. The first of these is perhaps the more common. It is the type of seeing with which, when presented an object, we immediately recognize and know what the object is. If I were to pull a pen from my pocket and say, "What is this object?", most people would instantly respond, "It is a pen." This is the type of vision which allows us to take a quick glance at a scene and

quickly reach a very good understanding of what is going on in the scene and the relationships among the principal components of the scene.

Once we have decided that the object we have in our hand is a pen we may then go to a second stage of seeing, where we determine the locations of the parts that make up the pen. This is the phase where we will discover that the pen has a cap instead of a push-point, and any other such significant details about the pen, which may be important in and of themselves, but do not detract from pen-ness. The search for these characteristics follows an overall comprehension of the "gestalt" of the object.²

Analytical vision, on the other hand, uses a more involved process to determine the contents of a scene. It appears to involve three processing modes which may be present in varying degrees, but which all work toward a common goal of scene analysis. To understand these modes we can use as a tool the somewhat arbitrary, abrupt division of tasks between brain halves which is found in current popular writings. The first mode is an inductive type of reasoning in which the various components of a scene are fitted together in a variety of ways in an attempt to build a coherent picture. It is useful to think of this as a left-brain process - logically oriented, very mechanical and ordered. Watching the efforts of this mode is another mode which works in much the same manner as the instantaneous vision. It sits there seemingly doing nothing until it reaches some type of instantaneous decision about what the left-brain process is assembling. It then jumps in and gives a complete response to the left-brain process. It is useful to view this as the right-brain process - somewhat irrational, intuitive, working in images. After the left-brain process has received an image from the right-brain process, it enters a third mode. In this mode it works very logically and deductively to try to prove or disprove the image given to it by the right-brain process.

²Much of this discussion has grown out of conversations with Kabrisky. For more information about two-stage vision see [Rab-
bit]. For more information about "gestalt" theory see [Rock].

This type of vision - intuit, conjecture, and prove - appears to have an external or adaptive control. If our vision system model is to be complete in its capacity to describe an arbitrary vision system it must then contain an analog to this control, which we can call desire. Desire seems to control both the threshold at which the right-brain process jumps in with an image, and the degree of precision with which that image will be tested by the left-brain process. A good example of this at work is cloud-watching. In the effort to locate an object in a cloud, the desire is high and so what would not normally match our vision as an elephant will trigger the right-brain to produce an elephant representation. The left-brain will follow this inaccurate identification with a fairly successful attempt to label any longish cloud as a trunk and any cloud billow as an ear. Desire appears to be controllable either internally by the visual system in response to its environment, or externally according to the wishes of the seer. It could even be viewed as a mechanism for inventiveness and creativity. Desire is also useful in understanding some non-representative works of art, although some art escapes even this mechanism.

2.4. External Control of Vision Systems

No vision system functions independently of its environment. There is always some type of external control system, which besides possibly influencing the desire of the vision system performs the more fundamental task of determining what tasks the vision system will perform. This external control may also determine whether the vision system is capable of performing that task, and if it is not, modify the capabilities of the system. The external control and modification mechanisms may be as simple as evolution, which through natural selection has created a broad variety of highly specialized vision systems, or it may be as complex as the human mind, with which humans define for themselves the vision tasks they wish to perform and provide themselves with adaptations to perform these tasks. Other forms of control include actively teaching

or training a specific animal to perform recognition tasks. While this involves no modification to the sensor, it can require a "rewiring" of the circuitry which generates an output.

It is important here to differentiate between output and response. The output from the vision may or may not be specifically the response of the animal being trained. In the case of a seeing-eye dog, where the dog becomes a replacement vision system for its master, the response of the animal is the vision system output. If on the other hand, we conditioned an animal to respond to a visual stimulus only when an additional stimulus (aural, touch, smell, etc) is present, we could clearly distinguish between the output of the vision system and the behavior of the animal. In fact, we would fully expect to be able to isolate and directly measure some such output in the nervous system of the experiment participant.

The reading of this text is an example of a task which humans have determined for themselves. Nature did not provide us with a system of writing and a set of letter interpreters in the human brain. Instead, we took the human vision system, which has a capacity for resolving fine detail and tuned it into a system, through the adaptation of symbols within the system capabilities, which can recognize letters and words. Of course, there are a number of people who have found their sensors inadequate for this task. They have therefore acquired a modification to give them satisfactory performance despite the system flaws. This modification usually takes the form of optical correctors, e.g. glasses or contacts. People have also added other adaptations to their sensor systems (microscopes, telescopes, I-R imagery, etc.), that have extended their capabilities. People have also developed mechanisms, such as computers, to extend the analysis capabilities of their vision systems. They may work by preprocessing inputs to the visual system such that they can be more readily understood (filtering, simplification, etc.), by accomplishing some task of the analysis portion of the visual system, such as attention direction, or in the extreme case by replacing the vision system. In almost all cases these mechanisms are directed at performing or improving some specific vision task.

3. Goals of a Vision System Model

If a general vision system model is to be developed, it must satisfy a number of specific goals. If the model is to be useful, it must be adaptable, perform its function better than a person does, according to some standard, it must be realizable, and it must be able to incorporate the advantages of both biological and technological methods. Further, any such model must consider the manner in which the information produced is going to be presented externally and must present the data in a useful fashion. On several of these points it becomes difficult to separate the model from a realization of the model, but without foundations in the model the goals become undoable in the realization.

3.1. Adaptability

Adaptability is important to a general vision system model in that the complete range of use of a system cannot be fully specified in advance of the use of that system. This is true for the simplest of systems as well as for those which will be asked to solve the most general of problems. Perhaps one of the simplest of vision systems, the bar code reader, provides an excellent demonstration of this principle in action. Designed to allow the quick check-out of groceries, the bar code reader has been adapted to everything from warehouse inventories and shipping labels to place-rankings of marathon runners. In this case, the simplicity of the concept of simply capturing a number has allowed flexibility in the way that number is processed. If the designers had chosen to assign patterns as product marks (i.e. this bar pattern is Cocoa Puffs, this is Fruit Loops) instead of assigning product codes and using patterns to represent code numbers, the system would not have had its ready extensibility into other areas.

The battlefield is another environment in which the requirements are subject to constant modification. If a weapons system incorporates a vision system which is trained to recognize aircraft, it must have a great deal of flexibility. A vision system which cannot respond to new or

unusual aircraft will be useless the first time it encounters a new-generation aircraft, or one which had not been considered during the system's design. If the vision system is to recognize hostile aircraft, it must be able to adapt to the current political situation. Aircraft recognition is only one of the areas of the battlefield which demand flexibility. Flexibility is needed to adapt to technological changes, to efforts to alter target signatures, and to overcome sensor and environmental degradation.

Adaptability is also important in allowing a vision system to mutate into a more advanced system. Rigidity of the vision system model means that any advancement or adaptation of the system must be done through a complete system redesign. This results in delays and increasing expense. It also inhibits the free flow of proven vision models and algorithms into new systems. Adaptability can also include the ability of the system to continue its mission through mutation. This may include a onetime mutation to react to a specific event, or a cyclic mutation to handle a periodic environmental occurrence.

3.2. An Improved System

There is not much point in creating a vision system which offers no improvement over human performance. Of course the measure of that improvement is somewhat relative. The ultimate vision system outperforms its human creators in every way. It can input a greater range of the spectrum; it can function in a greater range of illuminations; it can see farther; and it can detect finer detail. Not only will the ultimate system do all of this but it will perform tirelessly and at a lower cost than a human's. While this system is, as yet, available only on the Starship Enterprise, artificial vision systems can still outperform humans in a number of ways. They do not tire; they have perfect repeatability; and they can be made small and cheap. An additional benefit from the military perspective is that they are not only willing to sacrifice themselves freely, but there is also no one to bemoan the loss.

3.3. Realizable and Realistic

To be useful a vision system model must be realizable and realistic. It should not include any unrealistic assumptions. It should not require any processes or data which cannot be performed or provided. Any steps needed to reduce data to a form required by one of the constituent processes of the model must be provided for in the model. A model which performs vision tasks based on edges in an image must provide a means for obtaining those edges or it is incomplete. Further, if the model requires that those edges be pristine edges, with no discontinuities, distortions, or missing lines, the model is based on faulty assumptions and can be discarded as invalid, since such features cannot be realizably found in nature (except, of course, in bar codes).

To be realistic a vision system model must allow for and incorporate methods for overcoming noise both within the system and in its environment. The system must account for both random Gaussian-type noise, and for systematic noise. Noise is one of the constants of engineering, and proposed models which do not account for it are nothing more than toys.

A real-world problem, which can be thought of as a type of noise, is distortion of the input object. Distortion results from the fact that all objects, even of the same type, are not identical and that all sensors/environments do not have consistent characteristics over the field of view. Handwriting recognition is a common example of a pattern-recognition problem which must deal with a great deal of distortion. When we learn to read we must learn to deal with a wide variety of penmanship styles. We recognize as the same an 'A' which has a very wide base and a small cross bar, or an 'A' which has a narrow base and a broad slash across the center (Figure 4). It is almost as if the many styles of penmanship were designed to defeat the recognition system.

In the case of letters, it is perfectly acceptable for the recognition system to use for its output a pristine 'A', free of all distortions and discontinuities; however, there are other times when the output we want is not a pristine representation, but rather a clarified, but still distorted picture of the object in question.

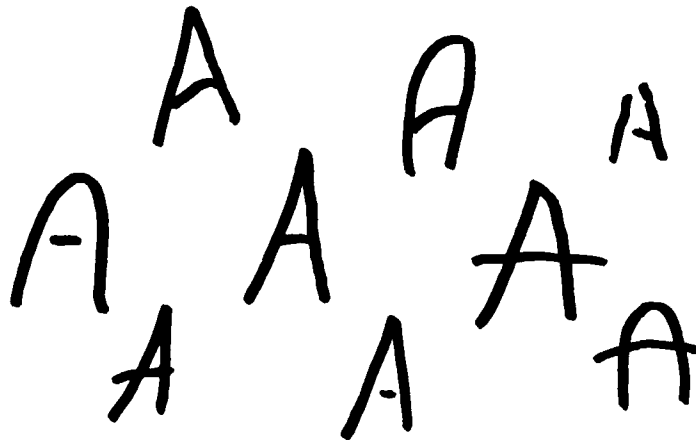


Figure A-4: Examples of the Written Letter 'A'

A fisheye lens gives an extreme example of the kind of distortion which may be added by a sensor, but this same type of distortion may also be found in less obvious places. Generally, we can expect to find some sort of distortion near the edges of a sensor's input. Distortion can also be caused by a strong light source which affects the function of the sensor, by magnetic fields or by a variety of other sources. Distortion may also be brought about by atmospheric conditions, by strong winds, or by other forces of nature. All of these factors conspire to make pattern recognition problems more difficult and show the need for any pattern recognition model to consider the effects of distortion.

Rotation is another problem that must be dealt with in systems which wish to function in the real world. There are two types of rotation which need to be considered. These are in-plane and out-of-plane rotation. In-plane rotation does not in itself obscure or alter shapes, edges or other features of an object. Because of this, in-plane rotations may be handled by a number of computationally intensive, but direct devices. Out-of-plane rotations can significantly alter the appearance of an object and compensation requires a great deal more consideration.

Any system must have a means of dealing with the scale of an object, whether it be normalizing all objects to a particular size either through internal manipulations, mechanical methods, or by the imposition of some type of external constraint on the scale of objects viewed by the system. Scale can also be used constructively within a system to distinguish between two objects.

3.4. Human/Machine Factors Considerations

The output of the vision system must be tailored to provide for the abilities of the user, whether the user of a vision system be machine or person. The presentation of the output needs to be concise, yet inclusive and formatted in a means which will enhance the user's ability to exploit the information. The most important information available is useless if it is not in a format the receiver can process.

4. References

- Block, Ned. "The Imagery Issue" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 577-607. (1990).
- Bechtel, William. "Connectionism and the Philosophy of Mind: An Overview" in *Mind and Cognition: A Reader*. Ed. William G. Lycan. Cambridge MA: Basil Blackwell. Pp. 252-273. (1990).
- Chase, William G. "Visual Information Processing" in *Handbook of Perception and Human Performance*. Vol II. Ed. Kenneth R. Boff et al. New York: Wiley & Sons. Section 28. (1986).
- Goldstein, E. Bruce. *Sensation and Perception, 3rd Ed.* Belmont, California: Wadsworth Publishing Company. (1989).
- Larson, Gary. *Farside*. Universal Syndicates. (1989).
- O'Hair, Mark Allen. "A Whole Word and Number Reading Machine Based on Two Dimensional Low Frequency Fourier Transforms." *PhD Dissertation, AFIT/DS/ENG/90D-1*, School of Engineering, Air Force Institute of Technology (AU). (December 1990).
- Rabbitt, Patrick. "The Control of Attention in Visual Search", *Varieties of Attention*. ed. Raja Parasuraman. New York: Academic Press. Pp. 273-291.(1984).

APPENDIX B

Testchip

1. Introduction

Testchip (Figure A-1) is a CMOS circuit designed and built as a MOSIS tiny chip. Testchip was designed to aid in the construction of ARES by providing a variety of known features on a

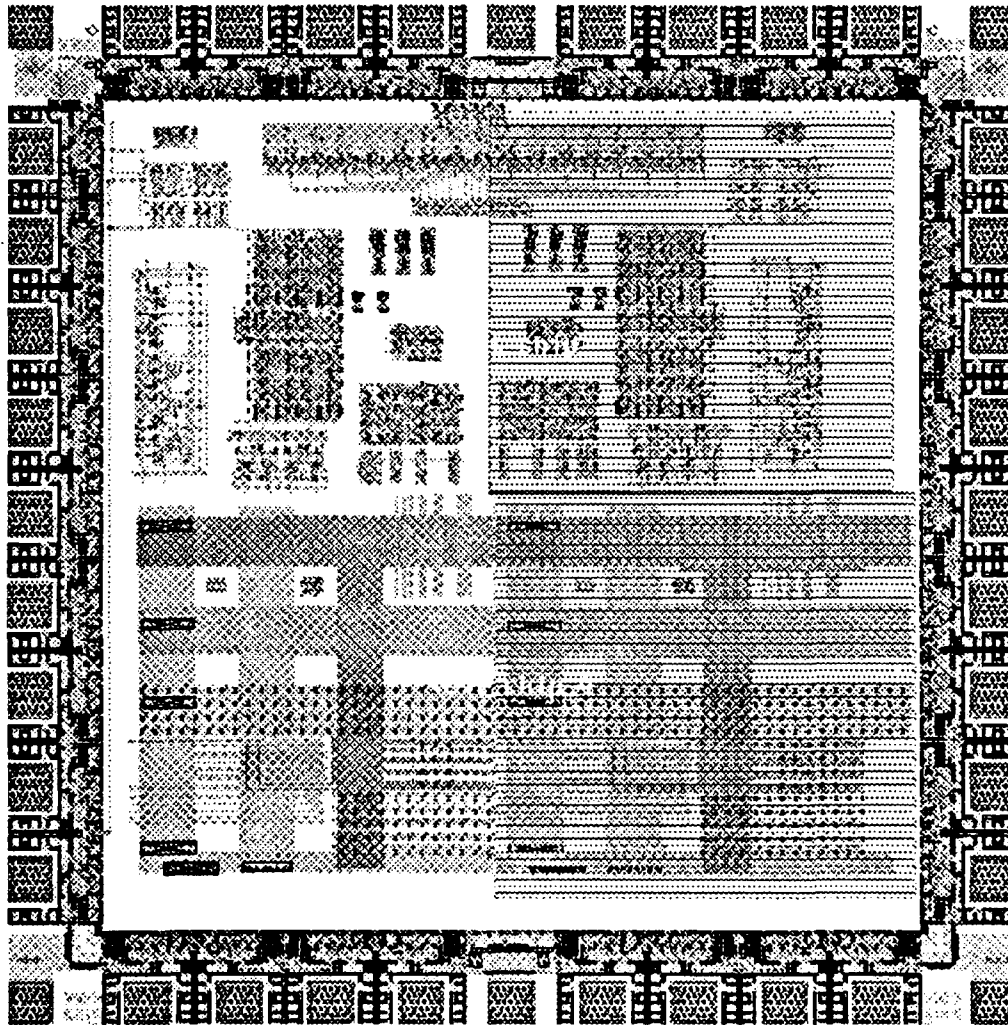


Figure B-1: Testchip

single chip which could be used for testing and tuning ARES components. Testchip was designed using MAGIC and fabricated through MOSIS. The four examples of the circuit fabricated each have two major sections, a circuit section and a large-feature section. Each section has two instantiations on the chips, one with passivation (over glass) and one without. This allows for the study of the effects of passivation on the processes used by ARES, and allows processes which may require the removal of passivation to be tested without the extra time and effort needed to remove passivation for test purposes.

2. The Circuit Section

The circuit section of Testchip contains a number of small circuits borrowed from other chips designed at AFIT. The objective of this section was to get representative samples from a number of different design styles. Some of the circuits are very compact and have many overlapping layers. Other circuits have a very loose style with lots of separation between components and very little overlap. A couple of circuit examples were chosen because of the large size of their features. Another set was chosen to be representative of some of the typical functional elements found on VLSI circuits. These include an inverter, an and gate and an adder. The circuits were not connected or wired together as the purpose of the chip is to study the circuits in static conditions, not under load.

Circuit sources include: PISO cell - Winograd Fourier Transfer Circuit [Shephard]. Multistage Shifter - Transcendental Function Generator [Dukes et al.]. Adder - Single Precision Multiplier [Jones and Gallagher]. Inverter - Unknown origin.

3. The Large-Feature Section

The large-feature section of the circuit was designed to study the effects of a single feature type or of a set of feature types consistent across a large area. This section was designed with

features large enough to cover the entire field of view of the microscope on maximum magnification. The large-feature area can also be viewed at a smaller magnification to provide samples of areas with more than one uniform feature. The views provided in this manner provide simpler patterns that are useful for the initial characterizations of many routines. A section of the large feature has a set of incrementally decreasingly sized features which are used to study the effects of size on the appearances of different materials.

4. Summary

Testchip has proved useful for characterizing portions of ARES. The large areas have helped to characterize the properties of lasers used for layer extraction and have allowed study of thin film dynamics for optical extractions. They have also proven useful for initial testing of edge finding and other image processing routines. The circuit portions of Testchip have provided a useful set of known circuits for testing extraction a variety of design styles. Testchip continues to serve these useful functions for further development of ARES.

5. References

- Dukes, M., E. Fretheim, and J. Kainec. "CORDIC Transcendental Function Generator Chip" *EENG 653 Class Project*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1987).
- Jones, Keith and David M. Gallagher. "Testing of the Floating Point Multiplier Chip" *EENG 795 Class Project*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1987).
- Shephard, Carl G. "Integration and Design for Testability of a High Speed Winograd Fourier Transform Processor." *MS Thesis, AFIT/GE/ENG/86D-46*. School of Engineering, Air Force Institute of Technology (AU). Wright-Patterson AFB OH. (December 1986).

APPENDIX C

The CHIP System User's Guide

THE CHIP SYSTEM

USER'S GUIDE

REVERSE ENGINEERING LABORATORY

AIR FORCE INSTITUTE OF TECHNOLOGY
Department of Electrical and Computer Engineering
Wright-Patterson AFB, Ohio 45433

1. Introduction

The CHIP system was designed as an extension of the Berkeley Computer Aided Design (CAD) tool MAGIC, to provide interfaces to AFIT developed image processing tools, the CLIPS expert system, the Rochester Connectionist Simulator (RCS), and drivers for the MITAS microscope stage controller. The extended system allows the use of all of these tools inside a CAD environment and provides the software foundation for the AFIT Reverse Engineering System (ARES). The interface to CHIP is built on the MAGIC interface and this manual assumes familiarity with MAGIC and its usages. The CHIP system interface was designed primarily to provide a method for researchers to reach into ARES and experiment during the design stages of the system. Therefore the interfaces are somewhat rough in areas, assume a large degree of knowledge of internal workings on the part of the user, and are limited in the assistance and error

checking they provide. Further, the CHIP interfaces are not stable as the system is still undergoing development. Within these limitations the CHIP system has proven a useful tool and can perform many valuable functions beyond those for which it was designed. A modified version of the CHIP system will eventually serve as the User Interface to ARES.

2. The Hardware Base

The basic CHIP system was designed to work on any platform which will support MAGIC, however it requires a large amount of working memory (24M minimum, 64M preferred). It has thus far been tested on Sun III and Sun IV work stations. The image processing subsystem is somewhat more restricted in that it requires pixrect libraries. Alternatives to using a Sun for these include using a non-Sun pixrect library or making some rather minor(though numerous) changes to the code. The image processing routines also require some type of framegrabber/framestore combination. The routines which access this are restricted to one portion of the code and can easily be updated to support any new hardware. Finally, many of the computationally intensive routines have been written for a specific set of vector processors. Several of these routines have non-equipment dependent duplicates which may take longer to process but produce the same results. The others can be replaced by creating new routines, either hardware independent or optimized for some other vector processor. The latter would be the preferred choice as many of these routines are very computationally intensive.

The machines which currently support the full capabilities of the CHIP system are Babbage - a Sun IV, and Mercury - a Sun III. Babbage contains a MAXVIDEO image capture and display system (Figure 1). The MAXVIDEO system supports up to 24 analog video inputs, including a 24bit RGB (Red-Green-Blue) capability. It also has a high speed digital camera input. There are three 512X512X8bit framestores and a 1024x1024x16bit region-of-interest store. The MAXVIDEO system can display RGB images or an 8bit colormapped image with overlays. The system

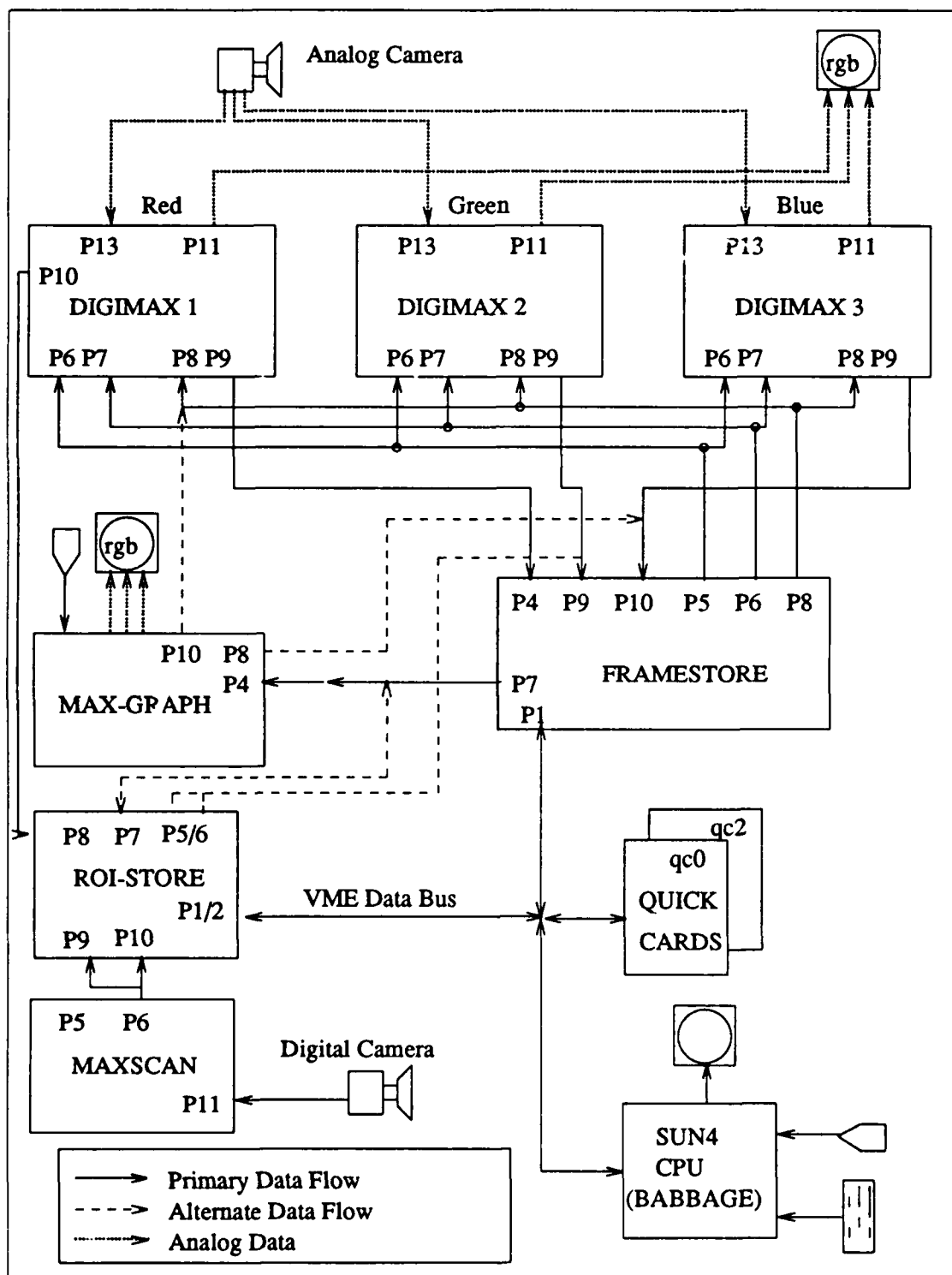


Figure C-1: The Maxvideo System

is highly flexible and readily user configurable. System setup and manipulation can be done using the "fsiTool", "ghiTool", and "dgiTool". Documentation for these is available in the Max-video manuals. The normal mode of operation is to have one RGB monitor displaying each of the three framestores and the other displaying the results of framestore '0' passed through the MaxGraph. The first monitor allows for a true color display for 3-space images. It can also be set to display either the camera inputs, or any one of the framestores. The second monitor allows for the use of overlays from the MaxGraph on a gray scale image.

Babbage also contains two 32-Mflop Quickcard vector processors. These vector processors allow for high speed processing of numerically intensive operations. Use of these is controlled by the program code. The MITAS controller is hooked to one port of Babbage, and an IBM PC-AT, which serves as a data capture and preprocessing device for laser and optical sensors, is tied to the other external port. Babbage is configured with 32M RAM and 64M of swap space.

Mercury has an ITEX FG-100 as its framegrabber/buffer. The FG-100 supports three video inputs. It has a 1024x1024x12bit framestore and can output any 512x480 pixel portion as either 8-bit greyscale or pseudocolor images. Information about the FG-100 is available from the FG-100 Users Manual. Mercury also contains two Quickcard vector processors and is configured with 8M RAM and 68M swap space.

3. System Startup

The CHIP system is started by invoking it using the command "chip". Any MAGIC command line arguments may be used with the chip command. To be able to invoke "chip" the user's path must include:

`/usr2/reverse/chip/bin`

The CHIP system can also be started using a Tooltool menu system. This menu provides not only the same functionality as the MAGIC Tooltool system, but also has buttons to activate the

special CHIP windows. The command is "tooltool -f chip.tt". Again any normal MAGIC command line arguments can be used.

Since CHIP requires large amounts of memory, it is suggested that users insure that the memory limits are set to their maximums. This can be done using the "limit" command either at the command prompt or in the user's .login or .cshrc file. As the system is largely experimental and does not always gracefully degrade (to put it nicely), it is also useful to use the "limit" command to set "coredumpsize" to 1K or some other low size which will prevent extraneous coredumps from cluttering disk space.

4. CHIP Windows

The CHIP system interface works through a number of windows which have been added to MAGIC. These windows can be invoked in the same manner as other MAGIC special windows through the use of the ":specialopen <window>" command. The windows added for CHIP include the CHIP window, the MITAS window, and the CLIPS window. In addition, an RCS window is available through a chip window command. Like MAGIC windows, each special window has a set of commands that are available when the cursor is placed in that window. As with other special windows the MAGIC global commands remain available in all CHIP windows, with the exception of the RCS window. The list commands available can be obtained by using the help command or from this manual. The commands for the CHIP menu will not be extensively listed by invoking the help command as the majority of these commands are invoked using the command chip and a special grammar. The RCS window usage is discussed in the RCS manual. Usages for other windows are discussed in the following sections.

4.1. The MITAS Window

The MITAS window (Figure 2) is used to send commands to the MITAS controller. The MITAS controller is used to move the microscope stage in the horizontal plane. The top bar of the MITAS window has the words "MITAS screen" and the current location of the stage in screen coordinates. The MITAS window uses three coordinate systems: stage, screen, and MAGIC. The first system is stage coordinates; these are the coordinates of the stage in motor step-size relative to the origin. Screen coordinates are based on the size of the video display output from the microscope. Screen coordinates count the number of screens the stage is moved from the origin. The transform from screen coordinates is some fixed multiple based on screen size. MAGIC coordinates are used for the MAGIC representation of the circuit. Stage coordinates are some fixed multiple of MAGIC coordinates based on feature size, plus an offset based on the relative

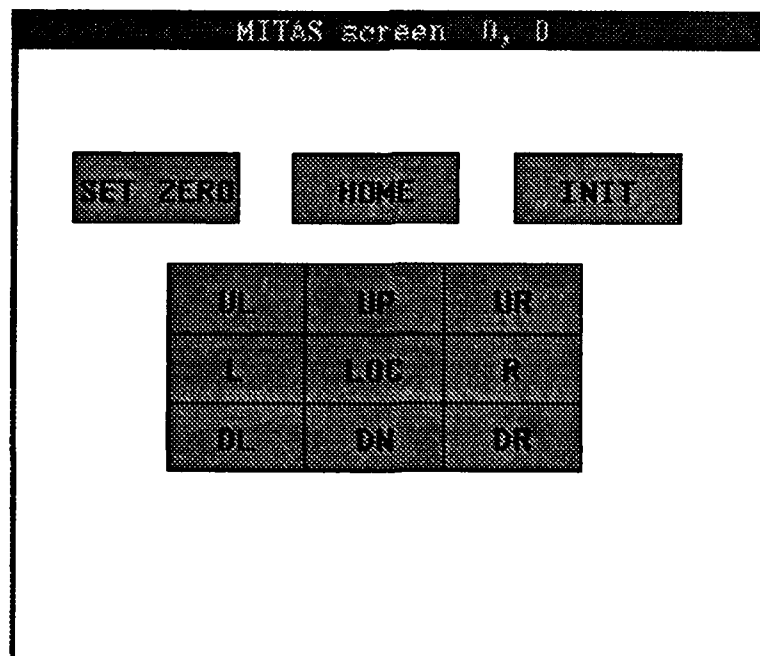


Figure C-2: The MITAS Window

locations of the origins. Before MAGIC coordinates can be transformed to stage or screen coordinates an association between coordinate-system origins must be established. This can be done by moving the stage to the MAGIC origin and setting that location as the stage origin. Before the other commands in the MITAS window are used the MITAS controller must be turned on and initialized. Once the controller has been turned on and initialized it can be accessed either by typing in commands or by using the window buttons.

4.1.1. Initializing the MITAS controller

Initialization of the MITAS controller requires a sequence of entries to be made on the front panel of the MITAS controller (Figure 3) after it has been powered on. Prior to power-on, the key switch of the MITAS controller should be in the program position. After power-on the LED

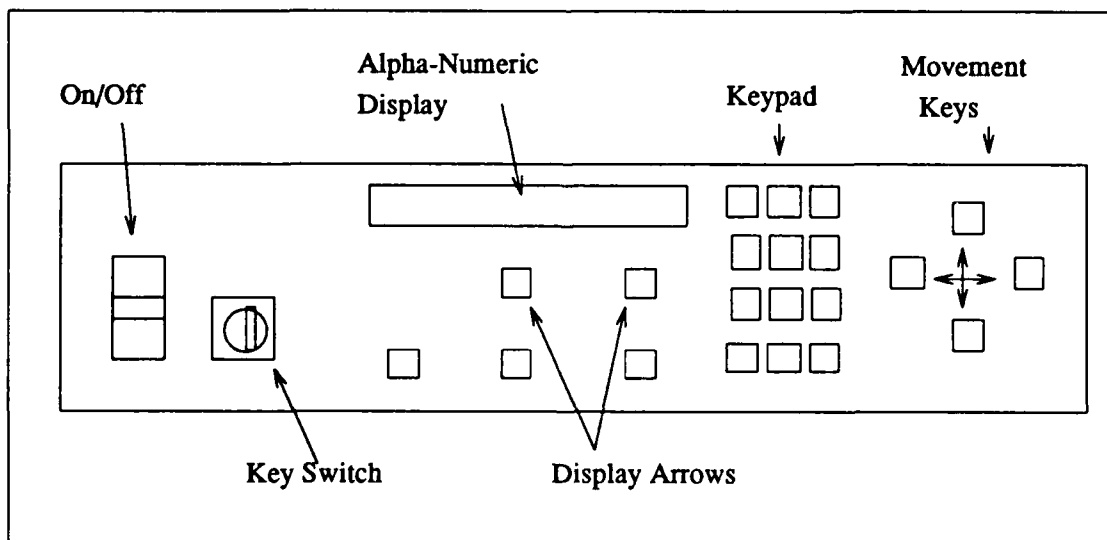


Figure C-3: The MITAS Controller

panel will go through a sequence of readings. The MITAS is ready for further initialization when the panel reads:

00 PGM_CTRL=5 0

At this point enter a 5 on the keypad. The panel will now read:

00 EX CTLS 3010

Enter right arrows until the display changes. Continue to enter right arrows until the display reads:

00 RS-232=2 0

Enter a two. Next enter a 4 to confirm RS-232 operation. The next option is to configure the baud rate for the RS-232 port. The computer port is set for 600 so enter a 0 rather than a 3 or a 1. The panel will now read:

BD 600=6 IK2=2 0

Enter a 6 and the MITAS controller will complete its set-up, confirming this with the entry:

RS-232 MODE

The MITAS controller is now ready for software initialization. This can be done either by clicking on the "INIT" button on the MITAS menu, or by entering the command ":init" while the cursor is in the MITAS window.

4.1.2. MITAS Buttons

The MITAS window has two sets of buttons. The first set, a row across the top, has an option (SET ZERO) to set the current stage location to coordinates 0,0. This command does not move the stage but rather changes the logical reference of the location. The middle button (HOME) moves the microscope stage to location 0,0. The third button is the INIT button discussed previously. Possible future options include a button to match the stage locations to the box location in the layout window and vice versa, as well as to establish and move to reference points and to execute command sets.

The center set of buttons is used for controlling stage movements. By clicking on one of the buttons the user can initiate stage movements relative to the button location (i.e. the upper-left button moves the stage up and to the left). The left mouse button moves the stage one step in the given direction. The middle mouse button moves the stage 10 steps in the given direction, and the right mouse button moves the stage one video screen size in that direction. The middle or "LOC" button on the screen gives the current stage location in screen and MITAS coordinates.

4.1.3. MITAS Commands

The commands of the MITAS window, which can be listed by using the help command with the cursor in the MITAS window, can be used to perform all of the functions available through the MITAS window buttons. There are also a number of commands which are not accessible through buttons. These include commands to use files, direct moves, parameter settings, and direct writes to the MITAS controller.

Movement commands for the MITAS can use either screen or stage coordinates. Commands that use screen coordinates are similar to associated commands using stage coordinates, but begin with the letter 's'. Later additions will include commands to use MAGIC coordinates for movements. Commands are ":move" or ":smove" for direct movements and ":offset" and

":soffset" for movements relative to the current location. The ":loc" command gives the current location in screen and stage coordinates. The ":zero", ":home" and ":init" commands perform the same functions as the corresponding buttons on the MITAS window. The ":size" command sets the size of a screen relative to the stage coordinates.

The ":load" command allows a command file to be read and directed to the MITAS controller. The command file should include each MITAS instruction (see the MITAS controller manual) on a separate line. The file is read and executed before any other commands can be entered. The ":mitas" command takes a string argument of MITAS-controller instructions and sends them directly to the controller for execution. The MITAS controller has a number of user-programmable parameters which control such things as motor speed, motor step-size, etc. Proper settings for these parameters have been established for the CHIP system; however, later experimentation may require that they be changed. The ":save" and ":reset" commands can be used to save and restore MITAS parameters to and from files. For the use of MITAS parameters see the MITAS controller manual.

4.2. The CLIPS Window

The CLIPS window (Figure 4) was created to allow the system user to interactively interface with the embedded CLIPS system. For specific information about CLIPS, the user is directed to the CLIPS User's Manual. The CLIPS window is invoked by the command ":spe clips". The bar at the top of the window lists the last rules-file loaded into CLIPS. CLIPS functions can be invoked either by menu buttons or command entry.

The CLIPS menu includes button commands for clearing, resetting and reloading rules into the CLIPS environment. The "reload" button causes the system to reload the file in the window title bar. The CLIPS window also has buttons to display CLIPS facts and the agenda. The "step" button is a special case of the CLIPS run function and causes the production system to fire one

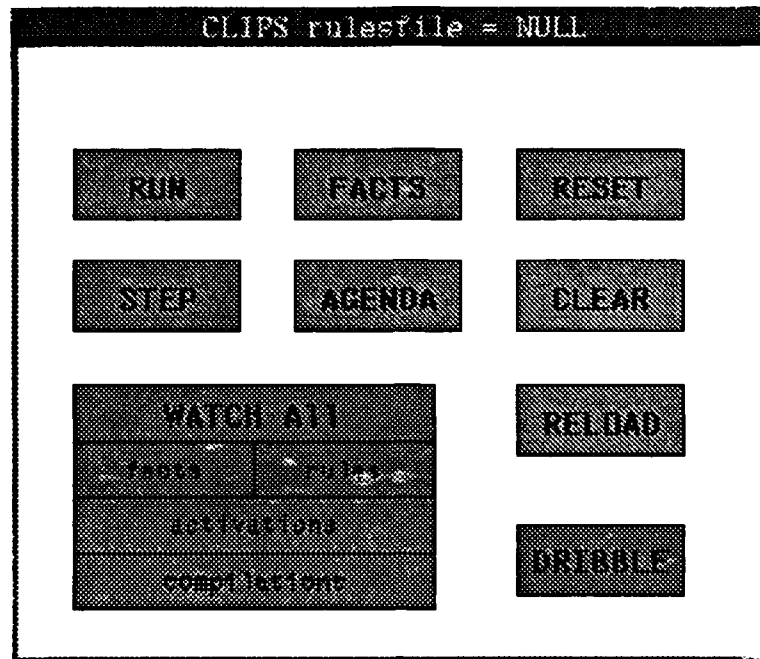


Figure C-4: The CLIPS Window

rule. The "run" button causes rules to fire until completion. The "dribble" button can be used to turn on and off the capture of CLIPS input/outputs to a file. The file used is the standard CLIPS default, "dribble.txt", or the last filename used with the "dribble" command. The "dribble" button changes color to show the user when the dribble option is activated. The "Watch" windows are used to toggle watching of the listed sections. The "Watch" windows also change color to indicate whether the function has been activated.

The CLIPS functions available from the command line are those listed in the CLIPS Advanced User's Manual for use with embedded CLIPS systems. They can also be found by using the ":help" command with the cursor in the CLIPS window. The commands allow for the control of CLIPS from the command line, as well as the assertion and retraction of facts. Facts can be retracted by fact number, which can be obtained from a listing by using the command ":facts".

255

4.3. The CHIP Window

The CHIP window (Figure 5) is used to control image processing functions of the CHIP system. The image processing section has a wide variety of autonomous functions which act upon a set of global memory pixrects. A memory pixrect is a storage space for an image in memory. These pixrects can have functions performed upon them, they can be written to a file for long term storage, or they can be altered by some functions. They operate as a kind of a picture register. There are three types of pixrects available with operations which can convert an image from one type to another. Each type of pixrect has one main named register upon which the majority of the functions perform their operations (see Table 1). There are also a number of supplementary named registers available for temporary storage and for functions which require the use of more than one image. The basic paradigm for image processing operations is to move

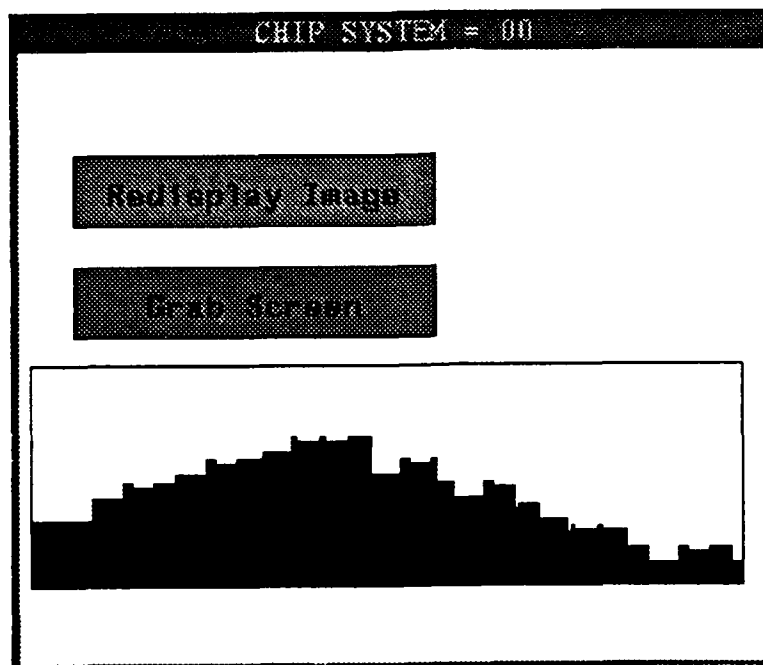


Figure C-5: The CHIP Window

Table C-1: Image Pixrects

Image Type	Primary Pixrect	Secondary Pixrects
8bit grey scale	SEARCH_RECT	TEMPLATE_RECT MASK_RECT LOGIC_RECT STORE_RECT
24bit 3 color	THREE_RECT	
float	DENSITY_RECT	
complex	D_FREQ_RECT	

the image into the proper pixrect, set any necessary parameters and then call the function to perform its operation.

The chip window is invoked by the command ":specialopen chip". Immediately upon startup of the chip window global variables are set to their initial values. Following this, the system will look for a ".chiprc" file. The system will look first in the current working directory, then in the user's home directory, and finally, if none has been found in either of these two directories, the system will look in the "chip" directory. If the system finds a ".chiprc" file it will execute any commands in the file. After completion of the initialization routines, the chip window will be displayed.

The window contains two sections: the button section and the histogram section. The button section has two buttons; the top one is used to grab a display into SEARCH_RECT. The bottom is used to display the contents of SEARCH_RECT onto the primary framestore. The screen-buttons are activated by clicks on a mouse button when the cursor is over the window button. The specific framebuffer with which the system interacts is controlled by the choice of mouse button. The left mouse button performs its actions on framebuffer 0 (or FG-100 equivalent), the middle mouse button uses framebuffer 1, and the right mouse button framebuffer 2.

The lower section of the CHIP window contains the histogram section. Upon window initialization the window contains a default histogram. This serves no other purpose than to look good and as a reminder of the function of the area. When the mouse is over the area the right button is used for computing histograms. The left and middle buttons are used to set the lower and upper boundaries respectively for histogram equalization. They can also be used to determine the value of a particular line on the histogram. This is displayed in the command window. If both the upper and lower boundaries of a histogram have been set, the right mouse button will initiate histogram equalization on SEARCH_RECT and then display the new histogram values.

4.3.1. CHIP Menu Commands

Commands on the CHIP menus are accessed by entering the command ":chip command-string" while the cursor is anywhere in the CHIP window. For example to read in an image the command would be ":chip f filename". The command string is derived from a grammar of menu options. When using the chip commands frequently it can be helpful to make use of MAGIC macros (see the MAGIC manual).

4.3.2. The CHIP Menu Grammar

The grammar used by the CHIP system is designed to be terse and utilitarian. It was designed to be easily parsed and to not interfere with the function, while permitting access to CHIP during ARES design. The grammar is passed to a set of simple parsers which act as menus. Letters are used in the grammar to represent commands. The commands are grouped onto several menus established for related functions. Each letter is followed by any needed parameters. Numerical parameters entered directly must be separated from each other by commas, as must string arguments, although when read from files they may be delimited by spaces. String arguments must also be separated from following commands by commas. Otherwise, the

use of commas or spaces as delimiters is optional, though often desirable to aid legibility for users. Any number of commands can be grouped together into a command string. However, the command string is limited to a total of 255 characters including delimiters and arguments.

Commands on menus other than the main menu can be called by first giving the command to switch to that menu. For example, if a user wanted to see the listing of 3-space parameters available on menu 4, the command would be ":chip bT". If a subsequent command in the same command string is from another menu, the command to switch to the current menu can be repeated and the parser will return to the main menu. Subsequent commands can then be processed in the same manner. To and together SEARCH_RECT and MASK_RECT and then display the results would require the system to go to menu 5; perform the command 'v'; return to the main menu and perform the command 'd'. The command would be entered as ":chip BvBd". At the end of every command line the parser will also return to the main menu. Thus the command sequence ":chip Bvd", would 'and' together SEARCH_RECT and MASK_RECT and then perform a "pre-QVA" function. The two commands ":chip Bv" and ":chip d" would perform the same 'and' operation, and then would display the resulting SEARCH_RECT.

Comments can be used in command lines entered either directly or from a file. Comments entered directly can be useful if the input is being captured to a file to help the user keep things clear. Comments entered from a file can be either enclosed in double quotes ("), or in the C-style bracket and star (/*). Comments entered directly must use the C-style option. Comments which are not closed by the user at the end of the command line will be closed by the system. The command ":chip d /* show an image */" would display an image and nothing else.

The letter 'Q' is used to leave the chipmenu system from any menu. In the current system it is not frequently used; however, when anticipated logical operators are added to the grammar, 'Q' will prove more useful.

4.3.3. Reading Chip Menu Commands From a File

The CHIP system has the capability of reading commands from a file. It can also store the pattern of commands given to it to a file. Reading is done using the command ":chip r filename". Writing is done using the command ":chip R filename". A second invocation of the command ":chip R" will stop the saving of CHIP commands. If the initial ":chip R" command is given as a part of a larger command string, that string will not be stored in the file. Likewise, a command string which includes a command to close a command file will be stored in its entirety in the command file. Therefore, it is generally a good practice to give these commands as single entries.

The file format for command files is an ascii representation with each command string given as a separate line. Each line can be up to 256 characters in length. The word "chip" is not needed as a preface to each line. A typical command file is given in Figure 6. Control will return to the command line after end of the file has been reached.

```
/* This command file combines */  
/* six Gabor transforms of an */  
/* image. */  
F  
h34w34r20o2  
k  
G0,1dlr45  
K  
G0,1dZlr70  
K  
G0,1dZlr110  
K  
G0,1dZlr135  
K  
G0,1dZlr160  
K  
G0,1dZdp17d
```

Figure C-6: Typical CHIP command file.

Comments can be added to a command file either using the C comment convention or by enclosing the desired comment in quotes. The characters inside the comment will not be interpreted as commands. Comments can be added to command files being interactively created by embedding comments into the command line entries. When doing this, comments should be enclosed in quotes. This will prevent MAGIC from removing the spaces in the comment. Ofcourseifyouaregoodatreadingrunoncommentsomitthequotes.

4.3.4. Image Display, Creation and Storage Commands

The command to move an 8-bit pixrect into a framebuffer is 'd'. This command can be supplemented with the number of the framebuffer in which the image is to be stored (0,1,2, or 3). The default is '0'. The DataCube system has three frame buffers available. To change the buffer being passed to the MAX-GRAPH start the fsiTool and enter "s(et) o(utput) #", where the portions of the command in parenthesis are emplaced by the program and # represents the number of the buffer to be sent to the MAX-GRAPH. 'q' will terminate the program. The framebuffer will now be displayed on the MAX-GRAPH, provided the overlay mode is enabled and the bits are not masked. On the ITEX FG-100 there is only one framebuffer available, however it can hold up to four 512X512 images. For the ITEX the framebuffer number is considered equivalent to the quadrant number (Figure 7). The quadrant being displayed can be adjusted using *toolbox*. If a display device is set to output that framebuffer, the image will be displayed. Otherwise, the framebuffer can be used as a temporary storage device. To retrieve an image from a framebuffer the command is 'F'. This command can also be supplemented with the number of the display buffer.

RGB images can be displayed on the DataCube system using the command 'i'. This will put the red, green and blue planes into the three framebuffers. They can then be displayed as a color image by putting each of the framebuffers to an A/D converter. Three-color images can be

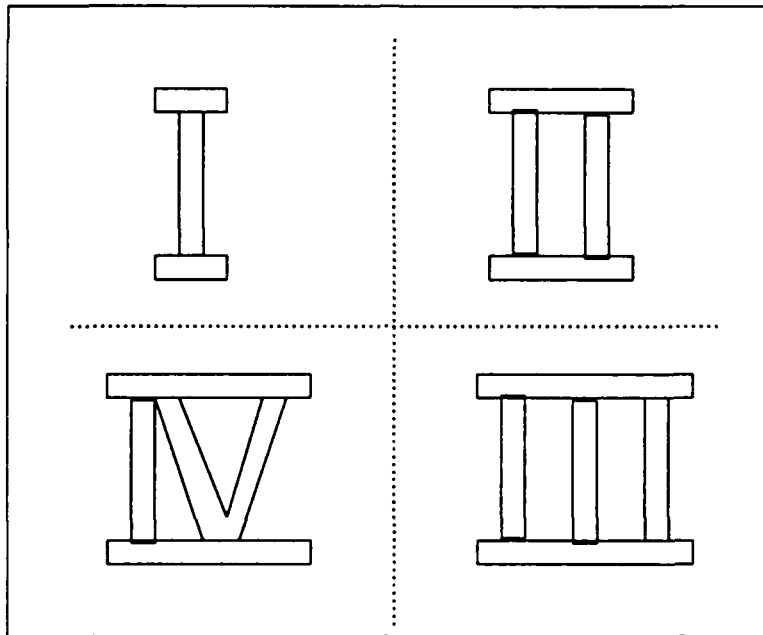


Figure C-7: Quadrants on the ITEX Framebuffer

retrieved from the framestore using the command 'I'. The framestore can also be used as a convenient means for moving a single plane of a three-color image to SEARCH_RECT for processing.

In addition to the standard display options, greyscale images can be displayed as overlay images on the MaxGraph. The command 'e' uses SEARCH_RECT as an overlay. The command 'E' uses MASK_RECT. For overlay usage refer to the MaxGraph user's manual.

Other special display options include the ability to display layer information from laser result arrays. The command is 'h <x_size> <y_size>'. The system can graph a line through SEARCH_RECT and display it as an overlay on the MaxGraph. The command is 'H <line_number>'; if no line number within limits is given, the command will prompt the user for a line to display. An ellipse can be displayed at any location that can be written to SEARCH_RECT by the command 'u'. Required parameters include the radius, the ratio of the a and b axes and the center location. The command 'U' will cause all points, with values above a threshold (TARGET_LEVEL), which are separated by a distance of X_SIZE (set in menu 6) to

232

be connected. The command 'v' will display a laterally inhibited line as an overlay on the Max-Graph.

Images can be stored to or read from a file. Images are stored in the image directory. The path for the image directory can be set using the 'f' command on menu 6. The full path should be given. The system default is to place images in and read images from the "data" subdirectory. The command to read 8-bit gray scale images is 'f filename'. This will read files in either the Sun rasterfile format, or the ITEX picture file format. The read routine will automatically detect the image type. To write an image to a file the command is 'F <type> filename'. The optional type designator can be '0' for Sun rasterfile or '1' for ITEX picture file. The default value is '0' - read a Sun rasterfile. Three-color images can be read and written using the commands 'j' and 'J'. Three-color images are stored only as 32-bit Sun rasterfiles.

Images can also be obtained by grabbing them using either the DataCube system or the ITEX FG-100. The command for grabbing single plane images is 'g'. There are a number of options available with this command. The option '0' will grab the number of frames indicated in a second option and average them together pixel by pixel. The default for this option is to grab a single frame. The option '1' will return the median value of the three framegrabbers. The second modifier for this option must be a multiple of three. If it is greater than three then the results from each framegrabber are averaged over the modulus of the number of frames to be used. Thus a choice of 'g 1 9' would return an image made up of the median values of three pixels averaged together within each frame. The option '2' does not grab a new frame but returns the current framestore contents to SEARCH_RECT. It is functionally equivalent to the command 'd'.

Three-space images can be grabbed by using the command 'G'. Options for this command currently implemented include '2' (which causes each of the framebuffers to grab an image and then stores the result in THREE_RECT); and '3' (which grabs an image one frame at a time from the first framebuffer).

Images can be moved between SEARCH_RECT and STORE_RECT using the commands 'k' and 'K'. They can be moved between SEARCH_RECT and MASK_RECT using the commands 'l' and 'L'. In both cases the lower-case commands copy the image from SEARCH_RECT and the upper case-letters copy the image to SEARCH_RECT. In menu 3 the commands 'k', 'K', 'l' and 'L' perform the same operations between DENSITY_RECT and the real and imaginary parts of D_FREQ_RECT respectively.

The command 'X' can be used to create a rectangle as a default scene in SEARCH_RECT. The options for this command include setting the starting location, setting the size of the rectangle, and setting the intensity of the rectangle. The default is to create a 10 by 10 rectangle at location 10,10 with an intensity of 100. The complete string is 'X <x_start>, <y_start>, <x_size>, <y_size>, <intensity>'. Parameters only need to be specified out to the last one which needs to differ from the default.

4.3.5. Template Manipulations

Many operations use a template to aid in their processing. The template can be of variable size up to 64 X 64. It is possible to store up to 5 templates at a time. Templates can be created from SEARCH_RECT using the command 'm'. In this case the location of the template can be decided using the cursors on the MaxGraph. This particular function is not yet implemented.

The command 'M filename' will cause a template file to be read into TEMPLATE_RECT. A subsequent call to the command 'n x y' will load a template from TEMPLATE_RECT with x and y as center coordinates. The size of the template loaded will be determined by the global variables TPLT_WIDTH and TPLT_HEIGHT. If there is no TEMPLATE_RECT, this command will cause the template to be acquired from SEARCH_RECT. A test template consisting of a small square in a dark field can be created by using the command 'N'.

Each template acquired is loaded into the next empty template position and the variable NUM_TPLTS is incremented by one. If all 5 template positions are filled, subsequent templates are loaded into the final position, wiping out whatever is there. The NUM_TPLTS can be reset using the menu6 command 'j #'. All templates in positions greater than the number entered will be lost, and subsequent templates will be loaded into the positions following that number.

4.3.6. Area Movements

Two commands have been made available to allow for movement of an area from one location on an image to another. These commands are 'w' and 'W' on the main menu. The first command moves a designated area to a new location on SEARCH_RECT. The second command moves a designated area from SEARCH_RECT to a designated area on MASK_RECT. The commands have six possible arguments. The first two designate the upper left coordinates of the area to be moved. The next two designate the size of the area to be moved. These four arguments are mandatory as the default area size is zero. The final two arguments are optional. They designate the starting coordinates for the area on the destination pixrect. The default is to place the area at the origin.

4.3.7. Layer and Block Extractions

One of the important tasks of ARES is to extract regions from a circuit and map them into MAGIC. This is done using the region extraction commands. There are a number of these commands. The command 'b', on menu 2, extracts a layer for an image and writes it to MAGIC. This command requires three arguments. The first is the expected intensity-value of the layer to be extracted. This can vary between 0 and 255. The second value determines how far to either side of the expected value intensities can vary and still be considered to be a part of the same region. The third argument is the short name for the tile type in which the region will be painted

in MAGIC. In addition to its arguments, the performance of the extraction routines can be affected by a number of global variables. In performing an extraction task the extraction routines begin near the upper left corner of the image. The exact location is determined by the global variables X_OFFSET and Y_OFFSET. At each point the routine looks at a two-lambda by two-lambda square (LAMBDA_SIZE). If the majority of the blocks in the square are within the correct intensity range the square is written into a reduced-size map. If more than 20% of the pixels are of the correct intensity range, a search is made to determine which adjacent pair contains the highest density of correct pixel-values. If this density exceeds 20% the one-by-two lambda region is written into the reduced-size image. This procedure is used because of the general assumption that features will be at least two lambda in size. After the entire image has been scanned a number of morphological operations are performed on the reduced size map to eliminate noise and fill possible gaps in the area. This map is then written into MAGIC at the proper location (X_START and Y_START determine the relative positioning of the particular image frame).

The command 'B' is used to segment three-space images into MAGIC. This works in the same manner as the single-space extractor, but uses the vector distance between the reference point and test points to determine which to accept. A new version of these extractors is being developed which uses store reference tables to determine the expected values and their standard deviations. These are created by using the 't' commands on menu 2. Finally, there is a two image extractor which uses the sum of the differences between the intensity values of a given point and the reference values.

The command 'c' on menu2, also works in the same manner as the regular single-image extractor, but its output is written to a CIF file. The CIF file can either be read into MAGIC, or used in some other fashion. One particular use of this version has been to place images on VLSI circuits sent for fabrication. This was done by seating the circuit designers in front of a camera

and grabbing a digitized image of their faces. These images were then sliced into intensity layers which were extracted to CIF files. The advantage of using CIF files was that the size of the picture could be adjusted by changing a parameter in the file. Once the pictures were adjusted to the correct size to fill the space available in the circuit, the pictures were edited in MAGIC to remove design rule errors. The results were quite impressive. A picture of the AFTT seal was also fabricated into the circuit.

The block extractors, 'C' and 'd', work from a binarized image. In this image, the edges of regions of contiguous common composition are portrayed in black. The regions themselves can

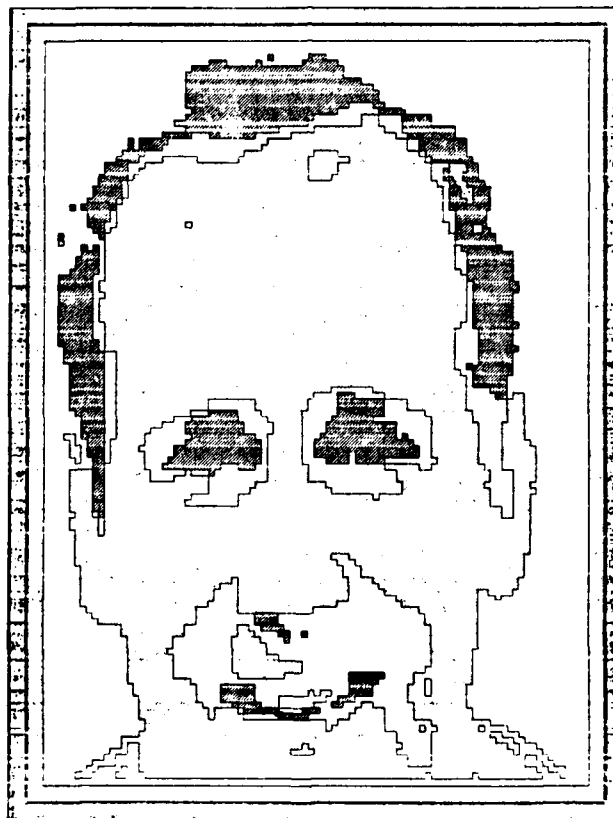


Figure C-8: Captain Linderman was in Control
(A Sub-Section of the Circuit)

be any color (128 works well). The binarized images are processed with a number of morphological operations which thin the lines and bridge any small gaps. Upon completion, a region-growing routine is used to find the bounds of each individual area. These areas are then extracted using the same methods as are used for the region-extraction routines. As each region is found, it is either written to its own unique subcell in MAGIC (case 'C') or to a CIF-style file (case 'd').

4.3.8. Laser Operations

Laser operations are used to extract information about the material composition of a circuit under test. The laser is read by a photometer connected to an IBM PC. The PC maintains tables which relate the readings to material types. As the PC makes new readings it determines the type of material at the reading location and assigns the reading a value based on the material type. The PC also determines the degree to which the assigned type fits into the particular material category. A decimal reading from 0 to .99 is added to the number to represent the closeness of the reading. The commands to support laser operations include 'F' on menu 2, which initializes the laser systems. Laser readings for a region are obtained using the command 'f' on menu 2. The outputs of these readings can be filtered using the command 'M' on menu 2. This causes a 3X3 modal filter to be applied to the laser results. To display these results the command is 'h' on menu 1. A test case can be generated for experimentation without having to actually turn on the lasers. This is 'z' on menu 1.

4.3.9. The RCS Simulator

The Rochester Connectionist Simulator (RCS) can be used to simulate all manner of neural and connectionist networks. The networks used in the simulator are programmed using C language constructs and are compiled into completed executable modules. The modules are then linked into CHIP. To use the modules the RCS controller section needs to be initialized. This is

done using the 'n' command on menu 2. Control is transferred to the RCS menu with the 'N' command. From this point the operations of the system are described in the RCS Manual [Godard]. The RCS has the ability to access all internal data within CHIP. Future extensions will allow a higher degree of control over RCS from the CHIP menus.

4.3.10. Neocognitrons

A neocognitron is a neural network which provides both recognition and reconstruction properties. The neocognitron works by examining a number of small areas and deciding which of them most closely approximates the set of objects for which it has a recognition capability. The result is passed to the next higher layer, which also examines spatially-offset groups to select its output. The final result is then passed back through the network to reconstruct the distorted version of the identified object. The particular networks implemented in this program are Pseudo-neocognitrons. Pseudo-neocognitrons maintain the neocognitron structure, but the internal equations have been modified to handle grey scale images. The commands for the neocognitron are on menu 2. The command 'w' initializes a pseudo-neocognitron. The command can either be entered with the name of a file which contains the network configuration data, or with no parameters specified. In the latter case, the program will query the user about the configuration. Figure 9 shows a portion of a typical pseudo-neocognitron configuration file.

The internal parameters of the pseudo-neocognitron can be tuned with the command 'T'. This will cause the program to query the user for values for particular constants. A particular constant can be left unchanged by pressing "enter" with no values given. The command 'w' reinitializes the pseudo-neocognitron. The command 'W' causes it to execute for the specified number of iterations. The default number of iterations (when no number is given) is one. Images can be loaded into the pseudo-neocognitron from SEARCH_RECT using the command 'v', or loaded from the pseudo-neocognitron into SEARCH_RECT with the command 'V'.

```

4   - four layer net
40 40 - input plane
4   - uc types in layer one
5 5 - first uc field
9 9 - uc spread
20 20 - us spread
10 10 - us field
0 0 1 - rotation, wave, freq.
5 5 - second uc field
.
.
.
5 5 - third uc field
.
.
.
5 5 - fourth uc field
9 9
20 20
10 10
135 0 1
4   - uc types in layer two
.
.
.
2   - types in layer three
.
.
.
60 0 1

```

Figure C-9: Excerpt from a Configuration File for a Three-Layer Pseudo-Neocognitron

There are a number of commands for displaying the internal states of the pseudo-neocognitron. The most general of these 'x' displays information about the basic structure of the pseudo-neocognitron - number of layers, number of cells of each type, etc. The coordinates for the locations of the Uc and Us cells can be obtained by the command 'X'. The input Us or Uc cell trees for a given layer will be displayed with the commands 'y <layer number>' and 'Y <layer number>' respectively. The values of the Us and Uc cells can be displayed by using the command 'z <layer number>' for the Us cells, and 'Z <layer number>' for the Uc cells.

Future expansion plans for pseudo-neocognitrons call for improvements to the basic algorithms themselves as well as a capability to incorporate multiple networks into a single version of CHIP. Currently, only one pseudo-neocognitron can be run at a time. In addition, as the pseudo-neocognitron uses large amounts of computational power, it is a good candidate for implementation as an independent process.

4.3.11. Density Domain Commands

The density domain is often a useful region for performing image processing tasks. This is, in fact, the domain into which the receptor cells of the eye cast their outputs. The density domain is, quite simply, a logarithmic mapping of the intensity domain in which we normally process images. To convert an image stored in SEARCH_RECT to the density domain requires the command 'a' on menu 3. This places a density representation of the image in DENSITY_RECT. The command 'A' converts an image from a density representation, in DENSITY_RECT, back into an intensity representation in SEARCH_RECT.

The density domain has available a number of processes which act in a manner consistent with those in the intensity domain, but which because of the domain changes may provide unique results. Among these are linear contrast enhancement (H), histograms (h), Gabor transforms (G), and logical operations. The logical operations are limited to those which deal with the values of the pixels rather than individual bits. This is because density domain images are represented by floating point numbers. Still, addition (w) and use of extreme value selections (z, Z) are possible.

One of the prime density domain operations is the FFT (f). This in conjunction with filtering and the inverse FFT (F), can be used to perform many interesting operations. One of these is the inverse visual filter (V); this filter de-convolves the processing done in the retina. The result is an image which represents the way we would see things if our eyes did no preprocessing. Of

course, we can also filter images to make them appear as we see them to prepare them for image processing (v).

There is also a variety of utility commands to move density pixrects about. These allow the user to look at either the real or imaginary portion of a FFT output, or to select some special input. This is an area which shows a lot of promise and should have many future additions.

4.3.12. Edge Detection

There are a variety of edge detection and related routines available. They are currently documented in the edge subdirectory. This documentation will be migrated to the User's Guide as the routines are adapted to the style of the remainder of the CHIP system.

4.3.13. External Programs

Menu 4 has commands which allow access to a number of external programs. These commands were initiated as a part of an experiment to determine the feasibility of using independent processes communicating through pipes. The command ' ', causes SEARCH_RECT to be piped to and displayed on the screen using *dsp*, a program from the ALV toolset. This command works, but has two major problems. First, the fork command causes the program memory requirements to double. This means that there is a strong possibility of running out of memory space. Second, the size of the images causes the pipes to become congested and the display time is extremely slow. Continuing research in this area may result in a more effective method.

4.3.14. Localized Transforms

Localized transforms are those transforms in which the value of each pixel is determined by its own value and those of a small neighborhood about the pixel. One such function is to place in each pixel the average of the pixels in a neighborhood about that pixel. This is done by the com-

mand 'a' on menu 5. The command must include the size of the area to be averaged. This number must be odd. A similar function subtracts the average of the area from the value of the center pixel. The resulting value is then multiplied by some twiddle factor and added to 127 (the middle of the intensity range). The result is an image which emphasizes regions with changing intensities (edges and such). The command, 'A' on menu 5, is given with the size of the area and the twiddle factor. A third local transform is a business operator. This uses a 3 by 3 region to make a determination of the local "business" of the area around each pixel. The command is 'b' on menu 5.

4.3.15. Global Transforms

Global transforms are those in which the output for each pixel is dependent on all of the pixels in the image. A good example of a global transform is the Fourier transform. Unfortunately, this transform is not available yet. A simple method to make it available would be to write a routine to move SEARCH_RECT into DENSITY_RECT without performing the normal conversions. Then the Fourier transform in the density domain routines would be available. Global transforms which are currently available include: the Gabor transform; linear contrast enhancement; and the Queen Victoria Algorithm.

Gabor transforms use TPLT_HEIGHT, TPLT_WIDTH and TPLT_ROT to determine the size and rotation of the Gaussian envelope. The sinewave modulation function rotates with the envelope with TPLT_FREQ cycles within the two-standard-deviation window of the envelope. The Gabor transform of SEARCH_RECT is taken at the command 'G' on menu 5. The first argument determines the phase of the sinewave modulation function. The second argument is for the decimation of the correlation function.

Linear contrast enhancement is used to improve the visual effect of an image. This routine functions much the same as the histogram equalization available from the display in the chip win-

dow. The major difference is that while the chip window function always puts the end values to 0 and 255, these values can be explicitly determined using the command 'h' on menu 5. The low and high break values need to be entered for each usage of the command. The low and high values are optional, and will default to 0 and 255 if not declared.

The Queen Victoria algorithm is a non-linear, non-causal operation which works wonders in cleaning up a noisy image. The algorithm smoothes areas to a single intensity value. Because of the method in which it does this, it may take several iterations for the process to stabilize. For this reason the command 'j', on menu 5, has the number of passes for its first argument. Experience has shown that '5' is a good number to start with. The next argument is the threshold which the algorithm will use to determine when it has encountered a new region. The choice for this number is highly dependent on the amount of noise in the image. A fairly clean image might use a number around 8-12, while a groddy image may require 20 or 30.

4.3.16. Logical Operations

There are a number of logical and related operations available on menu 5. These use simple bit-by-bit or pixel-by-pixel operations on two images - SEARCH_RECT and MASK_RECT - to produce a result. Table 2 lists these operations.

Table C-2: Logical and Related Operations

Command	Operation
t	Add
u	Highest absolute value
v	And
w	Or
W	Xor
T	Subtract
z	Modified And
Z	Highest value

4.3.17. Parameter Settings

All globally settable parameters are initialized at the start up of the CHIP window. They can be globally reinitialized at any time with the menu 6, 'c', command 'a'. A complete list of settable variables, their initial value, range, and menu entry is given in Table 3.

BoxPaintingCellName is the name of the root cell into which the layers being extracted are painted. It has as subcells the contiguous areas and is used by the expert system for its manipulations.

The TPLT variables are used for template operations such as the Gabor transform, averaging and other such operations.

X_PITCH and Y_PITCH are the number of pixels in the x and y directions in screen coordinates which map to one lambda in MAGIC coordinates. X_OFFSET and Y_OFFSET are the offsets needed to align the lambda grid with the image. The first line in the lambda grid should be X_OFFSET pixels from the origin.

TARGET_LEVEL is used by various image processing routines which need a designated pixel value in order to perform their processing. This value can be set by a number of routines, or from the menus. The setting should be immediately prior to the call which will use the value.

Table C-3: Chip System Defaults

NAME	MENU	DEFAULT VALUE	RANGE
BoxPaintingCellName	b	boxcell	alphanumeric
TPLT_HEIGHT	h	18	(4 - 66)
TPLT_WIDTH	H	18	(4 - 66)
TPLT_FREQ	i	1	> 0
TPLT_ROT	I	45	(0 - 360)
NUM_TPLTS	j	0	(0 - 5)
X_PITCH	p	6	(1 - 64)
Y_PITCH	P	6	(1 - 64)
X_OFFSET	r	1	(0 - 64)
Y_OFFSET	R	1	(0 - 64)
TARGET_LEVEL	s	200	(0 - 255)

4.3.18. Quick Reference for CHIP Menu Commands

```

/*****
* Program: chipmenu.c
*
* functions on this menu include:
*
*      Switch to other menus      (a - C)
*      Display and Read Images    (d - F)
*      Grab images                 (g - G)
*      Display and Read Images    (h - J)
*      Move Images                 (k - L)
*      Template Manipulations     (m - N)
*      Histograms                 (p - P)
*      Quit                       (q - Q)
*      Read from/save to file     (r - R)
*      Image adjustment           (s - T)
*      Special Displays           (u - V)
*      Create Test Cases          (x - X)
*
* Comments may be entered into Command files either as quotes or
* by using the 'C' language back-slash/star convention.
*
*****/

/* go to Second Menu */
case 'a' :

/* Go to menu 3 */
case 'A' :

/* Go to menu 4 */
case 'b' :

/* go to menu5 */
case 'B' :

/* go to menu6 */
case 'c' :

/* display on maxvideo */
case 'd':

/* Write to a raster file */
case 'D' :

/* Display SEARCH_RECT on MaxGraph */
case 'e' :

```

```

/* Use MASK_RECT as overlay on MaxGraph */
case 'E' :

/* Set SEARCH_RECT source to a raster file */
case 'f' :

/* Set SEARCH_RECT source to what is on the MAX-GRAPH or ITEX*/
case 'F' :

/* Grab a number of frames and combine them according to style, f s */
case 'g':

/* Grab a 3 plane image using grab option. */
case 'G':

/* Display the layer information from the laser */
case 'h':

/* Graph a line on the MaxGraph */
case 'H' :

/* Display a 24 bit pixrect */
case 'i' :

/* get THREE_RECT from the framestore */
case 'I' :

/* read THREE_RECT from a file */
case 'j':

/* write THREE_RECT to a file */
case 'J':

/* Move SEARCH_RECT to STORE_RECT */
case 'k' :

/* Move STORE_RECT to SEARCH_RECT */
case 'K' :

/* Move SEARCH_RECT to MASK_RECT */
case 'l' :

/* Move MASK_RECT to SEARCH_RECT */
case 'L' :

/* Take the template from the Max Graph */
case 'm' :

/* Take the template from a raster file */
case 'M' :

```

```

/* Get a template with x,y center. If another source hasn't
   been declared yet, use the standard source.          */
case 'n' :

/* Use a default template which is a square */
case 'N' : setstuff++;

/* Print Histogram of pixel values in SEARCH_RECT: pic size */
case 'p' :

/* quit doing nothing. */
case 'q':
case 'Q':

/* read commands from a file */
case 'r' :

/* Save all commands to a file */
case 'R' :

/* Overwrite bottom ~25 lines of screen as gray for printing */
case 's' :

/* Change pixel values from 0-255 to 0-128 */
case 'S' :

/* Shift SEARCH_RECT right by thresh pixels : thresh */
case 't' :

/* Print ellipse at location on SEARCH_RECT : radius,a,b,x,y */
case 'u' :

/* Cover area with lines lambda length between points above thresh */
case 'U' :

/* draw a lateral inhibited line */
case 'v':

/* Create a standard scene          */
case 'X' :

/* Create a phony laser results for testing display */
case 'z':

/* Handle comments in quotes - not valid through magic but ok
   from command files or direct */
case '""' :

/* Handle C type comments: */
case '/':

```

```

/*****
*
* Menu2 - Includes commands for:
*
*          layer and block extractions          (b - E)
*          laser system                          (f - H)
*          operations on laser results          (i - M)
*          RCS neural simulator                 (n - P)
*          neocognitrons                       (r - Z)
*
*****/

```

```

/* Return to first menu */
case 'a':

```

```

/* Extract a layer */
case 'b' :

```

```

/* extract a layer from a 3 space */
case 'B' :

```

```

/* Create a cif file of a layer, pres_val, slack */
case 'c' :

```

```

/* Create a blocks file */
case 'C':

```

```

/* Another Create blocks */
case 'd':

```

```

/* Get an array of laser measurements: x, dx, y, dy */
case 'f':

```

```

/* Initiate laser acquisition systems */
case 'F':

```

```

/* mode filter chip_laz using a 3x3 region; pixel being filtered */
/* is center pixel in 3x3 region; all the elements in chip_laz */
/* are processed. The input region size dx,dy must be manually */
/* scaled according to stepsize for the Mitas controller.      */
/* Current stepsize is set in the lay.h header file located in */
/* the layers directory.                                         */
case 'M':

```

```

*/

```

```

/* Initialize network simulator */
case 'n':

/* transfer control to the simulator */
case 'N':

/* reset a pseudoneocognitron */
case 't':

/* Tune pseudoneocognitron constants */
case 'T':

/* load image into a neocognitron from SEARCH_RECT */
case 'v' :

/* load the return image from a pseudoneocognitron to SEARCH_RECT */
case 'V':

/* Initialize a neocognitron */
case 'w' :

/* Run a neocognitron, number of iterations */
case 'W' :

/* Print info about a neocognitron */
case 'x' :

/* display the coordinates of uc and us cells for a neocognitron */
case 'X' :

/* display the s tree for a layer of a neocognitron */
case 'y' :

/* display the c tree for a layer of a neocognitron */
case 'Y' :

/* display the s values for a layer of a neocognitron */
case 'z':

/* display the c values for a layer of a neocognitron */
case 'Z':

```



```

/*****
*
* menu3.c - Menu of density domain operations.
*
*      conversion to/from density domain  - (b - B)
*      Clear/set image                    - (c - D)
*      Image transformations              - (e - G)
*      Histogram manipulations            - (h - J)
*      Image Movement and Storage         - (k - M)
*      localized manipulations             - (n - P)
*      filter operations                  - (r - V)
*      arithmetic and logical operations  - (w - Z)
*
*****/

```

```

/* Return to first menu */
case 'A':

```

```

/* Convert to density representation */
case 'b':

```

```

/* Convert density to intensity */
case 'B':

```

```

/* clear the imaginary part of D_FREQ_RECT */
case 'c':

```

```

/* create a default density rect */
case 'D':

```

```

/* Take fft of D_FREQ_RECT, and place power spect in DENSITY_RECT */
case 'f':

```

```

/* Take inverse fft of D_FREQ_RECT */
case 'F':

```

```

/* Gabor a density image ; wave_type, decimation */
case 'G':

```

```

/* histogram a density image */
case 'h':

```

```

/* Linear Contrast Enhancement of a density rect */
case 'H':

```

/* move a copy of DENSITY_RECT to the real space on D_FREQ_RECT */
case 'k':

/* move a copy back to DENSITY_RECT from D_FREQ_RECT real space */
case 'K':

/* move a copy of DENSITY_RECT to the imaginary space on D_FREQ_RECT */
case 'l':

/* move a copy back to DENSITY_RECT from D_FREQ_RECT imaginary space */
case 'L':

/* energy normalize (lambertize) in density space */
case 'n':

/* Perform a visual filter */
case 'v':

/* Perform an inverse visual filter */
case 'V':

/* Add density rect and D_FREQ_RECT real part */
case 'w':

/* Put the abs of the most extreme value into Drect, mode must be
calculated using dhist 'H' first */
case 'z':

/* Put the most extreme value of Drect and Dfreqrect into Drect,
mode must be calculated using dhist 'H' first */
case 'Z':

```

/*****
*
*           Routine: menu4.c
*
*           Image Noise Filtering Functions                      (a - A)
* [Edge Detection and Related Functions] (B - d)
* [Edge & Line Connection/Fill Functions] (D - E)
* [Region Extraction/Grouping Functions] (f - F)
* [Local Region/Neighborhood Statistical Functions] (g - T)
* Pipes to external programs          (r - Z)
*
*****/

/* Return to first (i.e. main ) menu */
case 'b':

/* 3x3 MEDIAN Filter SEARCH_RECT...no parameters */
case 'a':

/* GAUSSIAN Filter (SMOOTH) SEARCH_RECT image... */
/* Parameters: <window width, window hgt, std deviation>.. */
case 'A':

/* QUICKCARD CONVOLVE linear integer filter mask with SEARCH_RECT */
/* Parameters: <filter mask filename>.... */
case 'B':

/* Perform spatial GRADIENT MAGNITUDE of SEARCH_RECT image.. */
/* Parameter: <value to threshold edges on>.. */
case 'c':

/* Store GRADIENT ORIENTATION image of SEARCH_RECT in MASK_RECT.. */
/* Parameter: <orientation to display [0,45,90, or 135 degrees]>.. */
case 'C':

/* Generate single/multi-orientation GABOR image of SEARCH_RECT.. */
/* Parameters: <window hgt, window width, freq, std dev scale fac- */
/* tor, and orientation combination..combinations al- */
/* lowed are [1: 0 deg only, 2: 90 deg only, 3: 45 */
/* deg only, 4: 135 deg only, 5: 0 & 90 deg only, */
/* 6: 45 & 135 deg only, 7: 0, 45, 90, & 135 deg]... */
case 'd':

/* Connect edges using GRADIENT MAGNITUDE/DIRECTION information.. */
/* Parameters: <edge strength diff thresh, edge angle diff thresh> */
case 'D':

/* FILL 4-connected neighbor pixels of SEARCH_RECT pixel with a */
/* specific gray-scale value if their current intensity is w/i a */
/* +/- range of pixel (x,y).. */

```

```

/* Parameters: <x,y,intensity diff range,new gray value>... */
case 'e':

/* Connect lines in SEARCH_RECT image... */
/* Parameter: <threshold difference value> */
case 'E':

/* Quadtree SPLIT-AND-MERGE of SEARCH_RECT image... */
/* Parameters: <split-and-merge start level, intensity diff thresh>*/
case 'f':

/* Adjacency Grouping of split-and-merged SEARCH_RECT image... */
/* Parameter: <intensity difference threshold>... */
case 'F':

/* Display or Write statistics of an NxM even/odd region ... */
/* Parameters: <x,y,xsize,ysize,[display only (0) OR write only (1)]>*/
/* <x,y> coordinates of top left corner pixel.. */
/* DEFAULT operation is 0 (display only).. */
case 'g':

/* Display or Write Statistics of an NxM ODD NEIGHBORHOOD... */
/* Parameters: <x,y,xsize,ysize,[display only (0) OR write only (1)]>*/
/* <x,y> coordinates of center pixel.. */
/* DEFAULT operation is 0 (display only).. */
case 'G':

/* Display the histogram of an NxM even/odd region... */
/* Parameters: <x,y,xsize,ysize>.. */
/* <x,y> coordinates of top left pixel.. */
case 'h':

/* Generate mean and variances for layers in a three-space image.. */
/* Parameters: <x-start, y-start, area-size, layer-name>... */
case 't':

/* Print layer statistics table... */
case 'T':

/* ALV display... */
case 'x':
case 'X':

/* ALV glass.. */
case 'y':

```

```

/*****
*
* menu5.c - commands for:
*
*      Localized transformations (a - G)
*      Global transformations   (h - K)
*      Domain Transformations   (l - M)
*      Correlations             (n - S)
*      Logical Operations       (t - Z)
*
*****/

```

```

/* average the image: area size (must be odd) */
case 'a' :

```

```

/* Energy normalize the scene */
case 'A' :

```

```

/* business operator */
case 'b' :

```

```

/* Return to first (i.e. main ) menu */
case 'B':

```

```

/* normalize values in three space to sum to 100 */
case 'c':

```

```

/* Pre-Qva : area, twiddle */
case 'd' :

```

```

/* median filter SEARCH_RECT using a 3X3 region; pixel being */
/* filtered is center pixel in 3X3 region; ignores border pixels */
case 'D':

```

```

/* lateral inhibit a scene */
case 'e':

```

```

/* lateral inhibit a scene */
case 'E':

```

```

/* Relax a region of 250 around points above thresh; region, thresh */
case 'f' :

```

```

/* Process a gabor transform: wave_type, decimation */
case 'G' :

```

```

/* Perform contrast enhancement on SEARCH_RECT: low break,
   high break, low val, high val */
case 'h' :

```

```

/* QVA image in SEARCH_RECT : passes, threshold */
case 'j' :

/* Find the grid */
case 'M' :

/* Perform cross-check of image */
case 'n' :

/* add SEARCH_RECT and MASK_RECT */
case 't' :

/* Take the highest absolute value of two pixels as abs val */
case 'u' :

/* AND SEARCH_RECT and Mask_rect */
case 'v' :

/* OR SEARCH_RECT and Mask_rect */
case 'w' :

/* XOR SEARCH_RECT and Mask_rect */
case 'W' :

/* Perform correlation of SEARCH_RECT with quickcards */
case 'R' :

/* Subtract MASK_RECT from SEARCH_RECT */
case 'T' :

/* Perform correlation on SEARCH_RECT at points from MASK_
RECT */
case 's' :

/* Find centers of elliptic patterns in SEARCH_RECT
Input radius,a*10,b*10,thresh*10,picsize */
case 'O' :

/* Mask SEARCH_RECT (make 255) values HIGHER than logic_mask */
case 'y' :

/* Mask SEARCH_RECT (make 0) values LOWER than logic_mask */
case 'Y' :

/* AND (output is input if vals within thresh) of SEARCH and MASK */
case 'z' :

/* Modified AND (greatest abs val) of SEARCH_RECT and MASK_RECT */
case 'Z' :

```

```

/*****
*
* menu6.c - set parameters
*
*
*****/

```

```

/* set all defaults in one fell swoop */
case 'a':

```

```

/* Set box painting cell name */
case 'b':

```

```

/* Return to first (i.e. main ) menu */
case 'c':

```

```

/* Height for the template. */
case 'h' :

```

```

/* Width for the template. */
case 'H' :

```

```

/* Frequency (cycles per env) for Gabor */
case 'i' :

```

```

/* Rotation angle for Gabor envelope in degrees */
case 'I' :

```

```

/* Set number of templates equal to number */
case 'j' :

```

```

/* set X_PITCH */
case 'p' :

```

```

/* set Y_PITCH */
case 'P' :

```

```

/* set x offset */
case 'r' :

```

```

/* set y offset */
case 'R' :

```

```

/* set target level */
case 's' :

```

5. References:

- Everson, Phillip G. *ALV Public Domain Image Processing Toolkit for Sun Workstations*. University of Bristol: Bristol, United Kingdom. (February 1989).
- Artificial Intelligence Section. *CLIPS Reference Manual; Version 4.3 of CLIPS (JSC-22948)*. Lyndon B Johnson Space Center, Tx. (July 1989).
- CSP Inc. *QuickCard User Manual (ED-QC-004-04)*. Billerica, MA. (1989).
- Datacube, Inc. *Digimax Digitizer and Display Module (UM0041-6.1)*. Peabody, MA. (1988).
- Datacube, Inc. *Framestore Triple Frame Storage Module (UM0042-5.1)*. Peabody, MA. (1988).
- Datacube, Inc. *Max-Graph User's Manual; Revision 3.0 (UM0055-3.0)*. Peabody, MA. (1988).
- Datacube, Inc. *Max-Scan User's Manual; Revision 2.0 (UM0053-2.0)*. Peabody, MA. (1988).
- Datacube, Inc. *MaxWare Installation and Software Manual (UM0060-2.0)*. Peabody, MA. (1988).
- Datacube, Inc. *ROI-Store Region-of-Interest Storage Module (UM0054-2.0)*. Peabody, MA. (1988).
- Giarratano, Joseph C. *CLIPS User's Guide; Version 4.3 of CLIPS*. Lyndon B Johnson Space Center, Tx. (August 1989).
- Goddard, Nigel H., Kenton J. Lynne, and Toby Mintz. *Rochester Connectionist Simulator. Technical Report 233 (revised)*. The University of Rochester: Rochester, New York. (April 1988).
- Imaging Technology, Inc. *FG-100-V User's Manual (47-H10018-01)*. Woburn, MA. (1987).
- Imaging Technology, Inc. *ITEX-100 Programmer's Manual (47-S10008-02)*. Woburn, MA. (1987).
- Mueller, Michael R. *Investigation of Gabor Filters for Use in Reverse Engineering VLSI Circuits, AFIT/GE/ENG/89D-35*. Air Force Institute of Technology: Wright-Patterson AFB, OH (December 1989).
- Musciano, Chuck. *Tooltool User's Guide, Version 2.0*. Harris Corporation: Melbourne, FL. (1989).
- Scott, Walter S. et al. *1986 VLSI Tools: Still More Works by the Original Artists*. University of California at Berkeley: Berkeley, CA. (1986).
- Sun Microsystems, Inc. *Pixrect Reference Manual (800-1254-03)*. Mountain View, CA. (1986).
- Superior Electric Company, The. *Installation and Operating Instructions for Modulynx Motion Control Mitas Controller (Modulynx Intelligent Two-Axis System) Two-Axis Type PMS085-C2A and PMS085-C2AR*. Bristol, Connecticut. (1983).

APPENDIX D

The Chip System Programmer's Manual

THE CHIP SYSTEM PROGRAMMER'S MANUAL

REVERSE ENGINEERING LABORATORY

AIR FORCE INSTITUTE OF TECHNOLOGY
Department of Electrical and Computer Engineering
Wright-Patterson AFB, Ohio 45433

1. Introduction

The CHIP system was designed as an extension of the Berkeley Computer Aided Design (CAD) tool MAGIC, to provide interfaces to AFIT-developed image processing tools, the CLIPS expert system, the Rochester Connectionist Simulator (RCS), and drivers for the MITAS microscope stage controller. The extended system allows the use of all of these tools inside a CAD environment and provides the software foundation for the AFIT Reverse Engineering System (ARES). The interface to CHIP is built on the MAGIC interface and this manual assumes familiarity with MAGIC and its usages, as well as the CHIP User's Guide. The CHIP system interface was designed primarily to provide a method for researchers to reach into ARES and experiment during the design stages of the system. Therefore the interfaces are somewhat rough in areas, assume a large degree of knowledge of internal workings on the part of the user, and are

359

limited in the assistance and error checking they provide. Further, the CHIP interfaces are not stable as the system is still undergoing development. Within these limitations the CHIP system has proven a useful tool and can perform many valuable functions beyond those for which it was designed. A modified version of the CHIP system will serve as the User Interface to ARES. Programmers writing interfaces for the CHIP system need to be aware that their changes if not made correctly can have a serious effect on other users and programmers of the system.

2. The Hardware Base

The basic CHIP system was designed to work on any platform which will support MAGIC, however it requires a large amount of working memory (24M minimum, 64M preferred). It has thus far been tested on Sun III and Sun IV work stations. The image processing subsystem is somewhat more restricted in that it requires pixrect libraries. Alternatives to using a Sun for these include using a non-Sun pixrect library or making some rather minor(though numerous) changes to the code. The image processing routines also require some type of framegrabber/framestore combination. The routines which access this are restricted to one portion of the code and can easily be updated to support any new hardware. Finally, many of the computationally intensive routines have been written for a specific set of vector processors. Several of these routines have non-equipment dependent duplicates which may take longer to process but produce the same results. The others can be replaced by creating new routines, either hardware independent or optimized for some other vector processor. The latter would be the preferred choice as many of these routines are very computationally intensive.

The machines which currently support the full capabilities of the CHIP system are Babbage - a Sun IV, and Mercury - a Sun III. Babbage contains a MAXVIDEO image capture and display system. The MAXVIDEO system supports up to 24 analog video inputs, including a 24bit RGB capability. It also has a high speed digital camera input. There are three 512 X 512 X 8 bit

framestores and a 1024 X 1024 X 16bit region of interest store. The MAXVIDEO system can display RGB images or an 8-bit colormapped image with overlays. The system is highly flexible and readily user-configurable. Babbage also contains two 32-Mflop Quickcard vector processors. The MITAS controller is hooked to one port of Babbage; the PC-AT which servers as a data capture and preprocessing device for laser and optical sensors is tied to the other external port. Babbage is configured with 32M RAM and 64M of swap space.

Mercury has an ITEX FG-100 as its framegrabber/buffer. The FG-100 supports three video inputs. It has a 1024 X 1024 X 12bit framestore and can output any 512 X 480 pixel portion as either 8-bit grey scale or pseudocolor images. Mercury also contains two Quickcard vector processors and is configured with 8M RAM and 68M swap space.

3. The Programming Model

Routines for the CHIP system are written and tested in a different environment than that for system users'. This is done to prevent disturbances to the user environment before the changes are complete and their bugs ironed out. All program development is done in the "/usr2/reverse/work" directory subtree. Within this environment each developer has his own working copy of the system in the "work/bin" directory which he modifies as needed. When a section is completed it can be incorporated into the formal system.

The "work" subtree is divided into separate subdirectories which group together related routines. Each routine within the subdirectory is given a name with a standard prefix for that directory. This prevents routine-name collisions from occurring. The actual files in the subdirectory do not need to have prefixed names as they are used only locally within that directory. Each directory has a header file with the directory name and the appendix ".h". These header files are to be used for global data related to that directory and for external declarations of routines. They may also be used for a limited number of internal header functions.

The contents of each directory are compiled and loaded into a single object file. These object files are used when testing the routines under development. When the routines are completed the object files and header files are copied into the "lib" and "include" directories respectively. In this manner two people working on routines in separate subdirectories will not interfere with each other's development work. Periodically, the system manager will transfer the working portions of CHIP to the "chip" directory structure. This function is limited to one person to reduce chances for confusion and to provide control of the configuration management.

4. Makefile System

Makefiles are used to assist in compiling and linking routines for the CHIP system and to aid in controlling system management. There is a Makefile included in each subdirectory. Each Makefile has three sections. The first section is used to provide information to the compiler and linker on where libraries can be found, which compile time flags to use and what files in the directory are to be used. The next section gives any changes or special options needed for making on a sun3, or other specific machine rather than a sun4. The final section has specific instructions on how to make each target.

The first target in most Makefiles is "all". This target will establish the machine type on which the make is being run and then search for any files which have been changed since the last compile or which have been compiled for a different machine type. The Makefile will then compile all files it has found which need updating and load them into the subdirectory's master object file. The "all" target can be activated either by entering "make" or "make all". Specific pieces of code can be compiled by using the command "make filename (with no extension)". This, however, is not recommended as the same purpose can be accomplished by the target "all", and in directly making the target the make routine may miss some dependencies.

Once the master object file has been tested with the overall system it needs to be entered into the appropriate library. This can be done by the commands "make lib4" and "make lib3". The header file can be put in the include directory for general use by using the command "make inserthdrs". The "make clean" command will remove all object files and force a complete remake of the directory on the next "make all".

The Makefile for the chip subdirectory is somewhat different from the others. It includes named targets. For each of these named targets there is a variable list of object files. By adjusting this list of object modules, a system can be created which uses the object modules from the subdirectory under development rather than the library object modules. Each person working on the CHIP system has his own target. To tailor his copy of the system, the object module name he is working on is prepended with a redirection to the subdirectory. Thus "image.o" becomes "../image/image.o". After work on the module has been completed the list can be updated to indicate use of the copy of the object module in the libraries. The copy of the system which is made for the target is placed into the bin directory with the target as its name, except on sun3's where all targets currently make as "chip3".

The base directory Makefile has been set up to allow makes to be done in one command from this directory. For example if the developer "erik" has made modifications in the subdirectory "image" and now wants to test what he has done, he first checks to make sure that the object module in the "chip" directory has been redirected to the "image" subdirectory. This can be done with the command:

```
more ~reverse/work/chip/Makefile
```

The following lines of text appear in the file:

```
ERIK=edge.o menu.o xtct.o
      ../image/image.o video.o net.o stat.o
      clips.o dsp.o glass.o support.o layers.o neo.o
```

The particular portion which reads "../image/image.o", tells the make utility to use the object file in the image subdirectory, rather than the copy in the library. For the other object files "clips.o, layers.o, support.o, etc", the make utility uses the copies in the library directory (~reverse/work/lib4 or ~reverse/work/lib3). The user can then enter the command "make image erik" from the "work" or main directory. This will cause a "make all" in the image subdirectory, followed by a "make erik" in the "chip" subdirectory.

The base Makefile also has facilities for remaking the entire system and for installing the system in the "~reverse/chip" directory structure. In general these should be used only by the system manager (ie. keep your fingers off).

5. Adding Routines

The procedure for adding routines to the system begins with a search for a routine which can already perform the desired function. If this cannot be found then the developer should find the subdirectory with the most closely related routines. Subdirectories are group by function type, and by system requirements. Thus all routines which require use of the video input and display boards are grouped in the "video" subdirectory. Routines which perform functions related to edge finding and extraction are located in the "edge" subdirectory (see Table 1).

Once a directory has been selected the file can be created using the convention of beginning all names and global values with the subdirectory prefix. After the file has been entered the filename - minus the extension - is added to the CODE variable in the Makefile. After the code has been compiled with no errors the routine can be added as a menu selection (see "Adding to the Menus"). When the routine is added as a menu selection, an error-free copy of the object module is moved to the library. The object module does not need to have a properly functioning copy of the code, but it does need to properly compile so that others will not be unable to compile their versions of the system.

Table D-1: Subdirectory Groupings

Subdirectory	Contents
Bin	executables
Chip	main routine and header
Clips	Interfaces for CLIPS routines
Control	System control functions
Cortex	Cortex network routines
Data	test data and results
Doc	documentation
Edge	edge enhancement and detection
Finder	CLIPS rules for finding layers
Image	image processing algorithms
Include	header files
Lib3	Sun III object files
Lib4	Sun IV object files
Menus	menus
Neo	neocognitron code
Nets	RCS simulator models
Stat	image statistical information
Util	utility routines
Video	Image input-output
Windows	Window creation
Xtct	Region and layer extractors

The routine can now be tested and changes made without updating the library version of the module until the routine works correctly. If the code works correctly, but it does not do what you want it to, do not remove it. The code may prove valuable to someone else and save him countless hours of testing and development. Once a piece of code works a copy of the object module should be placed in the libraries and a description of the routine should be added to *The CHIP System User's Manual*.

When writing routines for the system the developer must exercise care to use the minimum number of global variables possible. The CHIP system is already extremely large and it needs to be kept down in size. Large arrays of the type needed for image processing can also consume lots of memory. To this end, a number of arrays have been globally declared in the "chip.h" header file. Whenever possible these arrays should be used rather than creating new arrays. If they

265

cannot be used the programmer should carefully consider both the use of dynamically allocated arrays and his approach to the problem.

As each file is written the programmer will find it necessary to add a number of header files to his programs. These should be limited to the minimum needed, but not at the expense of creating new global variables and definitions. Header files also need to be included in the proper order as some later headers redefine items from earlier headers. The proper ordering is first system headers (identified by their brackets < .h>), and then MAGIC header files (see `reverse/magic/include` for a list). The CHIP header files should be included last. Whenever MAGIC headers are used, dynamic memory allocation must be done using the `mallocMagic` system (see the MAGIC file "malloc.h").

All reads and writes to `stdio` should be done using the MAGIC commands `TxPrintf`, `TxGetLine` and `TxGetChar`. Descriptions of these can be found in the MAGIC "textio.h" file.

5.1. Hardware-Specific Programs

Hardware-specific routines and their specifications are available in the appropriate user's manuals. When coding a hardware-specific routine all calls, definitions, includes and global variables for that piece of hardware should be marked `"#ifdef"`. If practical a non-hardware specific version should be written for those machines which do not have the hardware available. Otherwise, flags should be left to tell the users that the function is not available on the machine they are using.

When using hardware functions it is important to include checks to make sure that the hardware is either in a known position, or to initialize it to a known setup. If this cannot be done or is not practical for the particular operation, be sure to document any steps which need to be taken prior to using the routine.

5.2. Adding to the Menus

Prior to adding an item to a menu, it is essential that there exists a routine in the libraries which will support the new addition. Once the item is in the library, the programmer should determine which menu has items most closely related to the new addition. The programmer also needs to check if there are letter designators available for that item.

Once the menu and letter designator have been determined the programmer determines the number and type of arguments which are needed. Arguments use common variables dependent upon the variable type. For example, the first integer variable used in each call will be 'i_a'. The routines *get_an_int*, *get_a_float*, and *getaname*, are used to collect the arguments and are then passed in to the routine. As little code as possible should be added to the menus. The purpose of the menus is to parse the command lines, not to run any image processing. For the most part any decisions which need to be made should either be made prior to entering the command, or be made by the called routine.

5.3. Adding to the User's Guide

After an item has been entered into a menu, a short reference should be added to the *Quick Reference* section of the *User's Guide* and a longer description should be added to the descriptive text of the *User's Guide*. The *Quick Reference* entry is most often the comment entered into the code at the "case" statement.

The entry into the descriptive text should include a description of what the routine does, any significant architectural features of the code, and suggestions on how the routine should be used. The description of what the code does needs to address all side effects of the code (pixrect updates, etc.) and what the routine can be expected to return. Suggestions on use of the code should address any limitations placed on arguments, should address their ordering, and should provide their suggested ranges of use. Any known errors and unsafe conditions need to be

specified. The description does not need to go into detail about how the code is written or to describe the algorithm in more than general terms unless that information is not available anywhere else, or understanding it is completely essential to use of the routine. In general, the reader of the manual is expected to be familiar with (or willing to become familiar with) normal pattern recognition/image processing techniques. Significant architectural features are those things which will make a difference in how the code is applied (for example, using space domain correlations as opposed to frequency domain correlations).

5.4. Adding to the Windows

Windows are programmed using MAGIC graphics. These graphics are not extensive in their capabilities; using them serves three purposes. First, the interface to MAGIC graphics is relatively simple. Second, there is no need to incorporate two graphics systems into the model. Trying to interpret mouse commands in several graphics systems would get really complicated. Finally, using MAGIC graphics allows portability to any system which runs MAGIC. The routines which define the windows are in the windows subdirectory. Each window has its own directory. In general each window subdirectory has a header file; a main file, which defines the configuration of the window, initialization and refresh operations; and a commands file, which defines the commands available in that window either by buttons or by command line entry. There is also an undo file which really does nothing but is required for MAGIC. Some windows also have additional files to perform utility functions.

When adding new windows to the MAGIC system, the easiest practice is to copy an existing window's files to a new directory and make incremental changes. To add a new window, one line must be added to the main.c file in the "magic/main" subdirectory. That line is a call to the *Init* routine for the window. The *Init* routine is then expected to add itself as a client to window server routines and do any initializations required for that window's services.

The MAGIC header files are currently the best source for locating calls and usages for programming in MAGIC windows. This may not be the most efficient method for doing things, but there is no Programmer's Manual available.

5.5. Adding CLIPS routines

Adding CLIPS routines is done by writing a C program to perform the desired function and by making an entry into the *usrfuncs()* routine in the *clips* subdirectory. The routines should be placed in the subdirectory which is most appropriate for the function they perform. In some cases this may be the *clips* subdirectory, but most often it is not.

A helper routine should be written for those routines which need to have arguments passed to them from CLIPS. The helper routines should be placed in the *clips* subdirectory. The helper routine will gather the necessary arguments and then make a call to the main routine. This allows the main routine to be written to be used from other C routines and not limited to calls from CLIPS.

5.5.1. CLIPS Rule Files

CLIPS rule files will be placed in the subdirectory appropriate to their function. CLIPS files should have a ".clp" extension to keep them from being confused with other file types. Several changes are expected with the next release of CLIPS. This will probably result in changes to the way things are currently done, so discussion of CLIPS is limited at this time.

6. References:

- Everson, Phillip G. *ALV Public Domain Image Processing Toolkit for Sun Workstations*. University of Bristol: Bristol, United Kingdom. (February 1989).
- Artificial Intelligence Section. *CLIPS Reference Manual; Version 4.3 of CLIPS (JSC-22948)*. Lyndon B Johnson Space Center, Tx. (July 1989).
- CSP Inc. *QuickCard User Manual (ED-QC-004-04)*. Billerica, MA. (1989).
- Datacube, Inc. *Digimax Digitizer and Display Module (UM0041-6.1)*. Peabody, MA. (1988).
- Datacube, Inc. *Framestore Triple Frame Storage Module (UM0042-5.1)*. Peabody, MA. (1988).
- Datacube, Inc. *Max-Graph User's Manual; Revision 3.0 (UM0055-3.0)*. Peabody, MA. (1988).
- Datacube, Inc. *Max-Scan User's Manual; Revision 2.0 (UM0053-2.0)*. Peabody, MA. (1988).
- Datacube, Inc. *MaxWare Installation and Software Manual (UM0060-2.0)*. Peabody, MA. (1988).
- Datacube, Inc. *ROI-Store Region-of-Interest Storage Module (UM0054-2.0)*. Peabody, MA. (1988).
- Giarratano, Joseph C. *CLIPS User's Guide; Version 4.3 of CLIPS*. Lyndon B Johnson Space Center, Tx. (August 1989).
- Goddard, Nigel H., Kenton J. Lynne, and Toby Mintz. *Rochester Connectionist Simulator. Technical Report 233 (revised)*. The University of Rochester: Rochester, New York. (April 1988).
- Imaging Technology, Inc. *FG-100-V User's Manual (47-H10018-01)*. Woburn, MA. (1987).
- Imaging Technology, Inc. *ITEX-100 Programmer's Manual (47-S10008-02)*. Woburn, MA. (1987).
- Mueller, Michael R. *Investigation of Gabor Filters for Use in Reverse Engineering VLSI Circuits, AFIT/GE/ENG/89D-35*. Air Force Institute of Technology: Wright-Patterson AFB, OH (December 1989).
- Musciano, Chuck. *Tooltool User's Guide, Version 2.0*. Harris Corporation: Melbourne, FL. (1989).
- Scott, Walter S. et al. *1986 VLSI Tools: Still More Works by the Original Artists*. University of California at Berkeley: Berkeley, CA. (1986).
- Sun Microsystems, Inc. *Pixrect Reference Manual (800-1254-03)*. Mountain View, CA. (1986).
- Superior Electric Company, The. *Installation and Operating Instructions for Modulynx Motion Control Mitas Controller (Modulynx Intelligent Two-Axis System) Two-Axis Type PMS085-C2A and PMS085-C2AR*. Bristol, Connecticut. (1983).

APPENDIX E

Subject Requirements and Consent Forms

1. Subject Requirements

Subjects must be able to see reasonably well, either unaided or using soft contact lenses, to participate in oculometer experiments. In particular, the following requirements must be met:

1. Must be 20/50 or correctable to 20/50 by soft contact lenses. This is determined through the use of a Snellen chart.

2. Near (20") unaided phoria (eso- or exo-)

Far (20') unaided phoria.

Acceptable: "ortho".

(If not "ortho", include next item.)

3. Compensating vergence test to the phoria.

(Base-in or Base-out)

Acceptable: at least 2 x phoria value.

4. Ophthalmoscopic Exam: KW#

Acceptable: KW# = 0, no retinal damage.

5. Slit Lamp Test (cornea, lens, and retina).

Acceptable: All refractive surfaces clear.

[see - *HMOF Facility Description*. Internal Memorandum. Helmet Mounted Oculometer Facility, Armstrong Medical Research Laboratory. Wright-Patterson AFB OH.]

211

2. Consent Form

I _____ am volunteering to participate in an oculometer experiment to study attentional mechanisms. No one has coerced me or intimidated me into participating in this program.

_____ has adequately answered any and all questions I have asked about this study, my participation and the procedures involved. I understand that the Principal Investigator or a designee will be available to answer any questions concerning procedures throughout this study. I understand if significant new findings develop during the course of this research which may relate to my decision to continue participation, I will be informed. I further understand that I may withdraw this consent at any time and discontinue further participation in this study. I also understand that the Medical consultant for this study may terminate my participation in this experiment if he/she feels this to be in my best interest. I will be required to undergo a preliminary eye examination and may be required to undergo further examinations, if in the opinion of the Medical Consultant, such examinations are necessary for my health and well being.

I understand I am entitled to no compensation for my participation in this experiment.

I understand that my participation in this study may be photographed, filmed or audio/video taped. I consent to the use of these media for training purposes and understand that the release of records of my participation in this study may only be disclosed according to federal law, including the Federal Privacy Act, 5 U.S.C. 552a, and its implementing regulations. This means personal information will not be released to an unauthorized source without my permission.

MY SIGNATURE INDICATES I AM DECIDING TO PARTICIPATE, HAVING READ THE INFORMATION PROVIDED ABOVE.

Volunteer Signature and SSN Date

Witness Signature Date

3. Addendum

ADDENDUM TO THE CONSENT FORM

In this experiment we will evaluate movements of the eye in viewing a variety of scenes of VLSI circuits and other objects. The observations will be used to evaluate attentional mechanisms. You will be required to view each scene and respond to verbal prompts.

Your participation will require two one-hour sessions. You must wear a specially designed helmet to permit eye position to be determined (cotton gloves must also be worn as a precautionary measure against visor damage). On the helmet are mounted 1) a dim source of infrared light and 2) a lightweight television camera. The reflection of the infrared light from the eye is monitored by a computer through the television camera. The amount of light used is less than that which would enter the eye while outside on a sunny day. This exposure amounts to less than one-half of the national safety standard. No physical, psychological, or social risks are expected by your involvement in this study.

No alternative means exist to obtain the required information.

If you have further questions later contact CPT Erik Fretheim (255-5276).

At your request, you will be given a copy of this form.

DATE

Volunteer's Initials